Multi-Agent Learning

Dominic Richards

St Peter's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary 2021

Abstract

Machine learning models are often trained on data stored across multiple computers connected by a network. Due to network stability, it is then often infeasible for a single central-hub computer to process and disseminate information. A solution to overcome this bottleneck is to consider a decentralised network akin to peer-to-peer and ad-hoc wireless networks. Namely, computers communicate to a subset of other computers at a time, with information then naturally propagating through the network.

This thesis investigates the *statistical performance* of models produced in such a decentralised framework. By modelling the network of computers as agents in a graph, we investigate two different statistical settings: *homogeneous*, when the data stored across the computers follows the *same* distribution; and *heterogeneous*, when the distributions are *different*.

In the homogeneous setting, and motivated by the problem of empirical risk minimisation, we consider the learning performance of a simple decentralised algorithm: *Distributed Gradient Descent*. Specifically, we demonstrate that guarantees on learning performance can be achieved through implicit regularisation alongside, in the case of non-parametric regression, a linear speed up in computational runtime for *any* network topology provided computers have a sufficient amount of data. In contrast, prior work has focused on *optimisation performance* through the more general *consensus optimisation* framework, which does not encode the finer statistical structure behind the scenes. More precisely, we demonstrate that this structure can be leveraged to both: allow model complexity to be controlled implicitly through algorithmic parameters; and that the information held by agents can be similar owing the phenomena of statistical concentration.

In the heterogeneous case a setting motivated by hyperspectral unmixing is considered. Specifically, we consider simultaneously recovering a collection of sparse signals (associated to agents), that are related in a manner reflecting the network topology. In short, the differences in the underlying distributions are encoded through a total variation penalty reflecting the network. Our approach then yields sample complexity savings over group lasso style methods when the signals are sufficiently related.

Contents

1 Introduction

Through the smart phones in our pockets, laptops in our homes and cars we drive, technology is now en-grained in nearly every aspect of our daily lives. Coupled with the latest technical advancements, these devices then continually collect and communicate information about their users and environment. This has resulted in a web of technological devices with the ability to adapt to our daily demands through the sharing and processing of information. Such a framework offers a different perspective on artificial intelligence from one of a sole single agent learning from its environment, to one now involving a collection of *distributed* agents intercommunicating with one another. Given the wide influence of such technological devices, it is therefore important to understand fundamentally *when* and *why* such a distributed approach to artificial intelligence is feasible and efficient.

The general artificial intelligence systems used in practice today typically follow an iterative process of learning. This is both a natural reflection of how humans learn as well as a computational convenience. Namely, while many of the artificial intelligence systems involve a complex pipeline of algorithms and processes, the key components are often constructed from a combination of simpler functions. This structure allows, with some high-school math (chain rule for differentiation), one to compute first order gradient information about the system parameters. Combined with simple greedy optimisation methods like gradient descent, this

allows parameters of the artificial intelligence system to be iteratively improved against some objective or loss. The process of computing the first order gradients in this manner is often referred to as *back-propogation* [1] (for a full history see also [2, Section 6.6]), and arguably lies at the heart of the successes of artificial intelligence used in practice today.

Following the success of iterative procedures in general artificial intelligence systems up-to now, it is reasonable to separate the learning procedure for a collection of agents, equivalently smart devices, into two iterative steps: *individual learning* and *shared learning*. In the *individual learning* step the agents simply learn from their environments as if they were not a part of a wider collection of agents. Meanwhile in the *shared learning* step, the agents share what they have learnt with other agents in the network. Reflecting how humans learn in practice, this provides a simple recipe for approaching the problem of learning with a collection of agents. While there is much variation in how each of the two steps can be performed and studied, we now proceed to introduce each of these steps in Sections 1.1 and 1.2 respectively. This will be followed by their combination which will be introduced in Section 1.3.

1.1 Individual Learning

Back-propogation has fueled the success of artificial intelligence systems as it provides a tractable approach to iteratively improve a wide range of machine learning systems. More precisely, its adoption has been driven by the intersection of three key areas: hardware technology, software technology and statistical methodologies. Specifically, hardware technologies such as graphics processing units (GPUs) have allowed practitioners to tackle larger problems as they efficiently perform computations resulting from back-propagation e.g. matrix-vector multiplications. Meanwhile, software frameworks like tensorflow (`https://www.tensorflow.org/`) and pytorch (`https://pytorch.org/`) have leveraged (through tools like Automatic Differentiation) the specific structure arising from performing back-propagation in order to develop easy to use computer programming languages for encoding an artificial intelligence system. With these areas responsible for *how* these systems are widely

adopted, *what* is being implemented and *why* it works is the focus of this thesis. In this regard, a number of advances, such as the simple functions to include in the neural network, have yielded large improvements in performance. For instance, convolution layers [3, 4] give improvements for problems involving images as they encode a natural form of translation invariance; recurrent neural networks [1] are well suited to problems with variable length inputs; and, more recently, attention and transformers architectures [5, 6] have allowed model's like BERT [7] and GPT-3 [8] to be pre-trained on ever-larger data sets.

While many of the reasons for why neural networks work well in practice follow a natural intuition, such as for convolution layers, it still remains unclear why, more generally, simple greedy learning methods built upon back-propagation work well across such a wide range of problems and system architectures. This puzzling problem has motivated a growing body of work in search for an underlying fundamental explanation. Indeed, some progress has been made in recent years. Specifically, the observation that overparameterised neural networks can generalise whilst interpolating [9] has yielded insights beyond the classical bias and variance trade off [10, 11]. Precisely in the case of ridge regression, it was then found that interpolating (zero ridge regularisation) was surprisingly optimal when the problem is sufficiently well-posed/easy [12, 13]. This being in contrast to previous work which focused on settings where *not* interpolating (non-zero ridge regularisation) was optimal [14]. Meanwhile, models like the neural tangent kernel [15], which exploit the structure arising from back-propagation, have shown that neural networks trained with back-propagation can be well-approximated by a linear model in certain regimes. Albeit recent work [16, 17] has suggested these regimes may not align with all the neural networks used in practice.

While a number of approaches are being developed and progress is being made, a fully general framework explaining the successes of modern artificial intelligence systems is currently ongoing research as of writing this thesis. Therefore, even though this question is not the primary focus of this thesis, since we consider applications involving such systems, the scope of problems we can consider is

currently limited to those we understand. Precisely, we limit ourselves to settings within the theoretical framework of *statistical learning theory* [18].

The specific statistical learning setting we consider assumes access to a loss function that measures the performance of a model on a specific datapoint. Given a collection of training data, the aim is to then produce a model that minimises the expected loss on new unseen data point i.e. that generalise from the training data. A simple approach in this case is *empirical risk minimisation*, which considers the model that minimises the average loss across the training data. While this approach is simple it comes with a number of drawbacks. Firstly, directly computing a minimiser can be unfeasible since the loss or model can involve a neural network, for example. Secondly, the training data is likely to contain idiosyncratic noise not useful for predicting new data points, and thus, cause some[1] models that directly minimise the loss to *overfit*. These two drawbacks can be overcome in a number of ways. Specifically, the problem can be made computationally tractable by considering specific models and loss functions. Meanwhile, the models ability to fit to noise, and thus overfit, can be restricted through regularisation. For instance in least squares regression, Tikhonov regularization [19] can be introduced to reduce the ℓ_2 norm of the regression coefficients.

In light of the previous paragraph, it is natural to view the simple greedy first order methods built upon back-propogation as introducing a form of *implicit regularisation* or *computational regularisation*. More generally, the two aforementioned drawbacks are overcome by *restricting to models that are computationally feasible*. To put it another way, the regularisation or model complexity is controlled by the limited computational resources at hand. This particularly convenient approach has roots in *early stopping* methods for inverse problems [19] and (through the aforementioned software and hardware technologies) has now been widely adopted beyond inverse problems to applications in artificial intelligence. Therefore, as we research the learning properties of distributed machine learning methods, it will be

[1]Note that some minimisers may generalise and be optimal. For instance, in overparameterised ridge regression, the interpolating least norm solution is attained when taking the ridge penalty to zero, and thus, is implicitly regularised by the squared norm.

fruitful to consider the setting of inverse problems since the *implicit regularisation* of iterative methods can be made precise here. More generally, we will see that using distributed approaches can introduce additional forms of implicit regularisation owing to, for instance, agents sharing information whilst they learn.

1.2 Shared Learning

The task of sharing information between a collection of agents can be framed in a number of ways, the specifics of which would depend upon the problem setting. Therefore, to be concrete, we follow a *distributed computation* viewpoint as if each piece of technology (car, phone, laptop, fridge etc.) were a computer. Pairs of computers can then communicate information to one another through either a local wireless connection, like Bluetooth, or the internet, in the case of the Internet of Things (IoT). This forms a network of computers, with the term *distributed computation* then being used as these computers can, in addition to communicate to one another, perform (potentially limited) computations on the information they have stored.

The motivation for utilising multiple computers in a network to perform computations over, say, a single computer is quite natural. For instance: redundancy, if we want to ensure a computation is completed given a computer breaks; security, if the problem contains sensitive information which must be stored separately e.g. names and dates of birth; computational speed, if we want to solve the problem quicker by having multiple computers work on it at once; or scale, if the problem is simply too large to fit on a single computer. Therefore, while we started with a specific setting, smart devices, this is just one instance of what is commonly referred to as *decentralised* computation. We now describe this more precisely.

How the computers are connected in a distributed computation system will naturally have an impact on the performance. Indeed, a vast number of protocols exist by which computers can communicate with one another. Therefore, for clarity, we focus on two types of networks: *centralised* and *decentralised*. Broadly speaking, computers connected in a *centralised* manner are all connected to a

common central-hub through which they can communicate to anyone else e.g. a star shaped network. Meanwhile, computers connected in a *decentralised* manner align with peer-to-peer networks like the internet, and thus, each computer can only communicate to a subset of any other computers. Naturally, each approach has its own advantages and disadvantages. In the centralised setting, information can propagate quickly between computers, but the central-hub is a bottleneck that, if broken or damaged, can disrupt or disconnect the entire network. Meanwhile for decentralised networks, information may propagate more slowly between computers as, for instance, the message may need to be routed through the network. On the other hand, the network is more robust since there is no single bottleneck which, if broken, brings down the entire network.

Returning to smart devices, the decentralised networks that appear in this case are connected in a certain way, or exhibit a certain type of *topology*. Specifically, devices often communicate with one another wirelessly, and therefore, can only communicate to a sub-set of neighbours within a certain distance or neighbourhood. This results in the network being viewed as forming a *constraint* on the communication between agents since information may not propagate quickly from, say, two computers either side of the network. That is, if one were to alternatively *design* the network for a data center you would ensure the distance between any two computers in the network is small so information can travel quickly across the network. In contrast, in wireless networks the devices are often spread over a wide area, and therefore, the distance between agents is often *forced* to be large. For example, random geometric graphs [20–22], which are a natural model for networks involving wireless devices, can often be modelled by a grid topology [23].

The communication *constraint* arising from the connectivity of the network has gained much attention within the distributed optimisation community, see for instance [24, 25] and references therein. The constraining effect can then be made precise in particular instances of the *consensus problem* where, in general, agents in the network must agree on some data value. The instance most relevant to this thesis then being the *distributed averaging* [26] problem, where each device holds a

number and wants to know (to a fixed precision) the average of all the numbers held by agents in the network. For the following discussion let us suppose each device is *synchronised* through, say, a clock, so the agents can then communicate to their neighbours in lock-step. One of the simplest algorithms to solve this problem then has each agent repeatedly average their number with the number held by their neighbours in the network. The speed at which the agents compute the network average (to a fixed precision) is then governed by the connectivity of the network which, in this case, can be controlled through the spectral properties of the network's Laplacian [27]. Specifically, the convergence of the iterative averaging is connected to the mixing time, or time to reach stationary distribution, of a random walk on the network. As a consequence, on poorly connected network topologies like grids and cycles, the performance of iterative averaging can scale unfavourably with the network size. This being due to a random walk exhibiting a "diffusive" behaviour [28] on these topologies resulting in a mixing time that scales with the network size to a polynomial power. In contrast, on well connected topologies like certain expander graphs, the mixing time, and thus number of iterations required, remains essentially constant as the network size grows.

Following the consensus problem literature, the view that the network topology can constrain communication has been adopted when considering *optimisation* problems that span the entire network. More precisely, given a certain structure of optimisation problem relevant for distributed machine learning (described in the following Section 1.3), recent hardness results have shown a slow down in computational performance [29, 30] *must* occur for decentralised algorithms on poorly connected network topologies like grids and cycles.

While the aforementioned hardness results do capture the most difficult setting, their applicability to artificial intelligence applications is questionable. Precisely, the hardness results exploit that the information held by each agent can be *arbitrarily* different. As we go on to discuss and noted within [31], applications in machine learning are often not in a regime where the information held by agents is arbitrarily different as, for instance, each agent may hold data sampled from *same* population.

This will allow us to overcome the hardness results in the case of artificial intelligence applications, and thus, offer a different view on the role of the network topology in decentralised machine learning.

1.3 Combining Individual and Shared Learning

Let us now consider a *homogeneous* distributed machine learning setting where a collection of agents both, exists in a general (possibly decentralised) network, and hold training data drawn independently and identically from the *same* fixed underlying distribution. This may arise artificially if a data set is split across a network of agents, or naturally, if each agent is associated with a device collecting data from the same population. Given this, each agent then wishes to use its data, and that of other agents in the network, to produce a model for prediction akin to the setting described in Section 1.1. We note, since all agents in the network hold data from the same distribution, they should be able to improve their prediction performance by communicating with one another.

Following the individual learning case, we begin with a *distributed empirical risk minimisation* setting where each agent considers the model that minimises the loss averaged across *all* of the training data within the network. This problem has then been studied previously through the more general *consensus optimisation* framework, see [32, 33] and **Decentralised Convex Optimisation** paragraph in Section 1.5, for instance. Precisely, this framework considers the optimisation problem of minimising an *average of functions* when each function is associated to, and can only be accessed by, a single agent in a network. *Distributed empirical risk minimisation* then fits into the framework of *consensus optimisation* when each function is simply the empirical risk, or average loss, of the data held by a single agent.

Naturally, due to the successes of machine learning, much recent work has gone into developing algorithms and theoretical guarantees for solving general *consensus optimisation* problems. As eluded to earlier, hardness results [29, 30] then show that solving general consensus optimisation problems with a decentralised algorithm *must* incur a computational slowdown for certain network topologies.

This is directly connected to the original *distributed averaging* consensus problem discussed in Section 1.2, in that, the slow down is associated to the mixing time of a random walk on the network topology.

While the slowdown for consensus optimisation is unavoidable as in the distributed averaging problem, the applicability to distributed empirical risk minimisation, and distributed machine learning in general, remains questionable. Namely, the functions associated to agents are *not* general, but actually averages of independent and identical realisation of a random phenomenon. For example, in the limit of infinite data, the functions held by each agent would concentrate around their expectation, and thus, be *identical* since they hold data sampled from the *same* population. Furthermore, framing the distributed machine learning problem as *empirical risk minimisation* problem fails to capture the fact that many machine learning problems exploit *implicit regularisation* techniques such as early stopping in order to avoid overfitting. These two observations leave open the natural question of whether: the statistical concentration of local quantities held by agents can be exploited to improve the computational performance of decentralised algorithms; and whether the implicit regularisation techniques for first order methods can be utilised in decentralised settings.

Following the *homogeneous* case, it is then natural to investigate approaches that are applicable when the data generating distributions of the agents are similar, but not identical. In the decentralised setting, this can then arise from the devices having different users and/or device specifications. In this case, the network connecting the agents can encode relevant statistical information about the data generating distributions. For instance, devices which are closely geographically may both: be able to communicate with one another; and have similar users and/or device specification. From a statistical perspective this *heterogeneous* setting can then be viewed as an instance of multi-task learning [34], and thus, it is natural that the differences in the underlying data generating distributions can be modelled using the graph connecting the agents. That is the graph now plays two roles: a computational role, encoding the communication channels between devices; and a

statistical role, encoding information about the relationship between the population distributions associated to the data held at each device. Following this, it is natural to develop statistical methodologies which leverage both of these properties. For instance, when the data at each device/agent is assumed to be produced from a well-specified linear model with a ground truth parameter, the ground truth parameter associated to each agent may vary *smoothly* across the network e.g. the differences of the ground truth parameters across edges in the graph has a small norm.

Given this setup, we now describe the remainder of the introduction. Section 1.4 summarises the primary contributions of the four works that make up this integrated thesis. Section 1.5 provides a summary of the related literature.

1.4 Contributions

This integrated thesis consists of four works applied to distributed machine learning. The first three works [35–37] investigate the questions set out in the previous section, namely, computational speed-ups and implicit regularisation for decentralised methods in the homogeneous setting. Meanwhile, the fourth work "Tree-Based Multi-Task Sparse Recovery with Total Variation Penalty" considers a heterogeneous setting where the data generating distributions of each agent are different. We therefore summarise the primary contributions by considering each area: homogeneous and heterogeneous, separately.

Homogeneous Setting As discussed previously and highlighted within [31], the homogeneous setting has previously been studied under the more general framework of consensus optimisation. This approach, while general, foregoes the fact that many machine learning models used in practice use *implicit regularisation* through early stopping with gradient decent; and that the functions associated to agents can be similar to one another owing to statistical concentration. To investigate these questions, the works in this thesis consider a simple distributed learning algorithm, *Distributed Gradient Descent* [38]. This algorithm, in short, has the agents alternate between an *individual learning* step where each agent performs a gradient descent step on the empirical loss of their own data; and a *shared learning* step where

agents average their model parameters with their neighbours in the network. Note the later step aligns with a single step of iterative averaging of model parameters in the context of the distributed averaging problem.

Given this setting, the first work [35] demonstrates for general classes of loss functions (smooth, Lipschitz, convex etc.) that optimal (in the minimax sense) statistical rates can be achieved by early stopping with Distributed Gradient Descent. These finding are of interest for two main reasons. Firstly, performing implicit regularisation allows one to avoid explicit regularisation like constraints and penalisation in order to achieve generalisation guarantees, and thus, allows for a simpler algorithm without projection steps and unbounded gradients. Secondly, the choice of communication protocol between the agents (as encoded by the communication matrix) has a regularization effect. In particular, the optimal choice of algorithmic parameters i.e. stepsize and number of iterations, was found to depend upon the inverse spectral gap of the communication matrix. In regards to utilising the similarity of the functions held by agents, it was found that the error can depend on a finer notion of gradient similarity between the agents compared to prior work. Precisely, the bound now depends upon the variance of the gradients and the deviation between the local objectives.

Building upon the first work, the second two works [36, 37] consider the specific case of non-parametric regression, and thus, align with the literature on inverse problems. In particular, in the second work [36], it is shown that when the sample size held by each agent is sufficiently large, that Distributed Gradient Descent can achieve a linear speed-up in computational time for *any* network topology. This arises precisely due to the statistical concentration of quantities held by agents in the network, as discussed previously. Note it also stands in contrast to the prior work on decentralised methods [29, 30, 33] which exhibit a computational slowdown for poorly connected network topologies like cycles. As far as we are aware, this is the first work in the context of machine learning to show an explicit speed-up in computational run-time for a decentralised algorithm with any network topology.

The third work [37] then builds upon the second [36] in two ways. Firstly, a refined analysis is performed that shows the speed-up holds under milder assumptions on the sample size held by each agent. Secondly, a practical algorithm is developed leveraging random features [39]. That is, in short, while the representer theorem [40] is typically used to parameterise functions for kernel methods, in the decentralised context this is not feasible as the data is split across the network. Random Features then offer a particularly convenient representation in this case as, both: the problem reduces to one of parametric regression; and the number of random features, and thus memory required, adapts to the difficulty of machine learning task.

Heterogeneous Setting The heterogeneous setting aligns with when the underlying data generating distributions between the agents are different, and thus, is a particular instance of multi-task learning [34]. Specifically, the differences between the underlying distributions have to be appropriately *modelled* to ensure that combining the information from each agent results in improved performance. How the information is combined then naturally depends upon the application in questions. Therefore, to be concrete, we consider a *sparse recovery* setting where each agent wishes to recover a sparse signal/parameter from a collection of potentially noisy measurements. It is assumed that the signals associated to each agent are related, with the manner in which they are related being encoded by the network topology connecting them. Namely, if an edge joins agents in the network then it is assumed that the *difference* between their signals is also sparse, with the sparsity of the differences being *smaller* than the sparsity of each individual signal. Such a setting arises within both: hyperspectral unmixing [41, 42], where pixels in an image are associated to sparse signals, and therefore, spatially correlated; as well as distributed machine learning [43, 44] when data is spread across a network that spans, say, a large geographic area, and thus, may contain a drifts in the underlying sampling distributions.

Given a collection of signals correlated in the manner just described, we investigate the sample efficiency of a method built upon *basis pursuit*, see for instance [45]. Basis pursuit, then being a method for recovering a *single* sparse

signal from a collection of linear measurements when the ambient dimension is larger than the number of samples. In short, basis pursuit aligns with finding a solution to a set of linear equations with the minimum ℓ_1 norm, and thus, is a convex relaxation of finding the minimum ℓ_0 solution.

The specific setting we then consider assumes the network aligns with a tree topology, since the case of a more general graph can be reduced to a spanning tree (if the signals are sparse with respect to a graph, they are sparse with respect to any spanning tree of that graph, see for instance [46]). Given this, the method sets out to recover a sparse solution for all of the agents simultaneously. This is achieved by finding a joint solution which has both small ℓ_1 norm at a root of the tree, and small *total variation* associated to the network. The *total variation* here being the sum of the ℓ_1 norm of the differences associated to edges in the graph. This penalisation follows the intuition that all of the signals in the network can be recovered in the following stepwise manner: recover the signal at the root; then recover the differences associated to edges in the network. Noting that the signal associated to any agent can be recovered by summing up the root signal and the differences along the edges going from the root to that agent.

Given this method, we provide guarantees on simultaneously recovering the support (set of co-ordinates with non-zero entries) of every agent's signal. In short, it suffices for the sample size for the non-root agents to grow with the sparsity of the *differences* multiplied by the square of the number of agents. In contrast, all prior work (to the best our knowledge) for joint recovery requires the non-root agents to have their sample size scale with the sparsity of their signals (this is required in order to satisfy an incoherence condition). Our approach then offers sample complexity savings when the sparsity of the differences are small relative to the sparsity of the underlying signals i.e. the signals across agents are sufficiently related.

1.5 Related Literature

The work in thesis spans a number of topics. For clarity the literature associated to each topic will be introduced within its own paragraph.

Distributed Averaging Problem One of the earliest references on the Distributed Averaging Problem is [26], with now a large body of work investigating different protocols, see [25, 28, 47] and references there in. In the setting we consider, the collection of numbers held by the agents can be represented as a vector, with the iterative local averaging of the agents encoded as repeatedly applying a row-stochastic matrix (rows sum to one) to this vector. Note the matrix then aligns with the transition probabilities of a Markov Chain on the network with uniform stationary distribution. Therefore, the convergence of the numbers held by each agent to the network average is connected to the rate at which this Markov Chain convergences to its stationary distribution, and thus, the spectrum of the matrix. In the case of a symmetric matrix (reversible Markov Chain) the convergence speed is specifically linked to the spectral gap i.e. difference between the largest and second largest Eigenvalues of the matrix [27, 48]. In short, for grid or cycle topology this then leads to slow convergence owing to a symmetric random walk exhibiting *diffusive* behaviour, see for instance [28]. Much work has then been motivated to design matrices, as well as more sophisticated protocols, that exploit the structure of certain network topologies to yield faster convergence [23, 49–54]. Some of these methods then include both lifted Markov Chains [28, 55] as well as higher order polynomials of the matrix which can then be computed iteratively [52, 54].

Decentralised Convex Optimisation Many problems in decentralised multi-agent optimisation can be phrased as a form of consensus optimisation, and thus, its study has a long line of literature [26, 32, 33, 56–64]. The observation that many distributed machine learning problems, in particular, can be phrased as a consensus optimisation problem has motivated a number of recent works developing algorithms under a variety of different assumptions. We highlight a few key works. The work [38] introduces the Distributed Gradient Descent algorithm that we study. The original algorithm was applied to constrained optimisation problems, and thus, included a projection step that made the analysis more technical. These technicalities were over come by the dual method within [33], ultimately providing a clean dependence on the network topology through the communication matrix's inverse spectral gap.

Owing to the fundamental connection to distributed averaging, the inverse spectral gap dependence has then gained much attention with, in particular, [29, 30] showing the dependence is unavoidable for general consensus optimisation problems i.e. lower bounds on oracle complexity. We also note a popular algorithm for solving variants of consensus optimisation problems is the Alternating Direction Method of Multiplers (ADMM), see for instance [62]. Finally, in contrast to the decentralised of this thesis, a large number of works have leveraged the statistical setting of distributed empirical risk problems in centralised settings [65–73]. The algorithms and analysis in these settings rely on a central-hub node to collect and disseminate information, and therefore, cannot be easily extended to the decentralised setting.

Implicit Regularisation of Gradient Descent with General Losses As stated previously, a number of works in thesis adopt the statistical learning theory [18] framework to study the learning performance of first order gradient based methods. Analysis of the generalisation performance for these methods then typically falls into the single pass or multi-pass setting. In the single-pass setting, each data point is used once to obtain an unbiased estimate of the expected out of sample or test loss, and thus, generalisation guarantees directly follow from optimisation guarantees for stochastic gradient descent [74]. Meanwhile in the multi-pass, setting each data point is used multiple times, and thus, the gradients are not guaranteed to be an unbiased estimate of the expected loss. This motivates the approach of [75] which decomposes the test error into optimisation and generalisation errors. Following this approach for the distributed setting, the optimisation error can then be bounded using techniques in decentralised convex optimisation [33, 38, 74] (see also above). Meanwhile, one of the primary contributions of this thesis is to study the generalisation error in the decentralised context. To do so, we utilise the technique of stability [76, 77], which has been previously applied to gradient descent for both convex and non-convex losses [78–82]. Notably within our work it is found that the bound on the expected generalisation error does not depend upon the network topology for Distributed Gradient Descent.

Implicit Regularisation with Gradient Descent for Non-parametric Regression As eluded to previously, implicit regularisation by early stopping with iterative first order gradient based methods can be traced back to the inverse problems literature [19, 83]. The wide adoption of first order methods in machine learning applications has spurred research into studying the prediction performance of iterative methods for inverse problems. Some of the earlier works then include [84–86], which have been extended in a number of settings to include: multi-pass stochastic gradient descent [87]; accelerated gradient methods [88]; stochastic gradient descent with large step sizes and averaging [89–92]; stochastic gradient descent with random features [93]; centralised divide and conquer schemes [67, 70, 71]; and pre-conditioned gradient descent [94].

Joint Recovery of Sparse Signals One of the chapters within this thesis considers joint recovery of sparse signals, and therefore, we introduce some literature in this area. The problem of recovering a single sparse signal from linear measurements has gained much attention within the field of compressed sensing, see for instance [45]. Most notably, the sensing/design matrix satisfying a *Restricted Null Space* is both a necessary and sufficient condition for recovering a sparse signal with basis pursuit. A sufficient condition for the sensing matrix to satisfy the Restricted Null Space Property is that it satisfies a *Restricted Isometry Property* up to the sparsity level of the signal [95]. This can then be satisfied for a variety of random matrices provided the sample size is larger (up to logarithmic and constant factors) than the signal sparsity. Naturally, a large body of works have then investigated simultaneously recovering a collection of sparse signals [96–103]. The majority of these methods consider group lasso style penalties that jointly penalise the signals across the tasks. In short, this postulates that, if the signals are stacked to be rows in a matrix, then this matrix should have a block-sparse structure. To then achieve theoretical guarantees for these group lasso style methods, the sensing/design matrices associated with each task are typically assumed to satisfy an *incoherence condition* [101–103]. Similar to the Restricted Isometry Property, this then requires the sample size of each task to grow with the sparsity of the underlying signal.

This means the agents are unable to use fewer samples than the sparsity of their underlying signals, even if all of their signals were identical. In contrast, for the total variation approach we consider, the sample size for all but one of the tasks grows with the sparsity of the signal *differences*, and thus, can be smaller than the sparsity of the underlying signal when the signals are sufficiently related to one another.

1.6 Thesis Structure

The remainder of this thesis is an integrated format, and therefore, each chapter is associated to a manuscript. The manuscripts and their associated chapters are summarised as follows.

- **Chapter 2**: "Graph-Dependent Implicit Regularisation for Distributed Stochastic Subgradient Descent", **Dominic Richards**, Patrick Rebeschini. In *Journal of Machine Learning Research*, 2020.
- **Chapter 3**: "Optimal Statistical Rates for Decentralised Non-Parametric Regression with Linear Speed-Up", **Dominic Richards**, Patrick Rebeschini. In *Advances in Neural Information Processing Systems*, 2019.
- **Chapter 4**: "Decentralised Learning with Random Features and Distributed Gradient Descent", **Dominic Richards**, Patrick Rebeschini and Lorenzo Rosasco. In *International Conference on Machine Learning*, 2020.
- **Chapter 5**: "Tree-Based Multi-Task Sparse Recovery with Total Variation Penalty", **Dominic Richards**, Sahand Negahban, Patrick Rebeschini. In *Preprint*, 2020.

2

Graph-Dependent Implicit Regularisation for Distributed Stochastic Subgradient Descent

Journal of Machine Learning Research 21 (2020) 1-44 Submitted 9/18; Revised 7/19; Published 1/20

Graph-Dependent Implicit Regularisation for Distributed Stochastic Subgradient Descent

Dominic Richards DOMINIC.RICHARDS@SPC.OX.AC.UK
Department of Statistics
University of Oxford
24-29 St Giles', Oxford, OX1 3LB

Patrick Rebeschini PATRICK.REBESCHINI@STATS.OX.AC.UK
Department of Statistics
University of Oxford
24-29 St Giles', Oxford, OX1 3LB

Editor: Sathiya Keerthi

Abstract

We propose graph-dependent implicit regularisation strategies for synchronised distributed stochastic subgradient descent (Distributed SGD) for convex problems in multi-agent learning. Under the standard assumptions of convexity, Lipschitz continuity, and smoothness, we establish statistical learning rates that retain, up to logarithmic terms, single-machine serial statistical guarantees through implicit regularisation (step size tuning and early stopping) with appropriate dependence on the graph topology. Our approach avoids the need for explicit regularisation in decentralised learning problems, such as adding constraints to the empirical risk minimisation rule. Particularly for distributed methods, the use of implicit regularisation allows the algorithm to remain simple, without projections or dual methods. To prove our results, we establish graph-independent generalisation bounds for Distributed SGD that match the single-machine serial SGD setting (using algorithmic stability), and we establish graph-dependent optimisation bounds that are of independent interest. We present numerical experiments to show that the qualitative nature of the upper bounds we derive can be representative of real behaviours.

Keywords: Distributed machine learning, implicit regularisation, generalisation bounds, algorithmic stability, multi-agent optimisation.

1. Introduction

In machine learning, a canonical setting involves assuming that training data is made of independent samples from a certain unknown distribution, and the goal is to construct a model that can perform well on new unseen data from the same distribution (Vapnik, 1995). Given a certain loss function that measures the performance of a model against an individual data point, the classical framework of regularised empirical risk minimisation involves looking for the model that minimises the empirical risk, i.e., the average loss over the training set, under some notions of regularisation, and investigating the performance of this model on the expected risk or Test Risk, i.e., on the expected value of the loss function taken with respect to a new data point.

In the distributed setting, data is stored and processed in different locations by different agents. Each agent is represented by a node in a graph, and synchronised communication is allowed between neighbouring agents in this graph. In the decentralised setting typical of peer-to-peer networks,

there is no central authority that can aggregate information from all the nodes and coordinate the distribution of computations. In sensor networks, for instance, data is collected on different sensors and each sensor communicates with nearby sensors by sharing model parameters. In the setting where the distributed data is assumed to be generated from the same unknown distribution, the goal is to design iterative algorithms so that agents can leverage local exchange of information to learn models that have better prediction capabilities as compared to the models they would obtain by only using the data they own.

In recent years, primarily due to the explosion in the size of modern data sets, the decentralised nature in which modern data is collected, and the rise of distributed computing platforms, the setting of distributed machine learning has received increased attention. From an optimisation point of view, problems in decentralised multi-agent learning are typically treated as a particular instance of consensus optimisation, and a variety of techniques have been developed to address this general framework, starting from the early work of Tsitsiklis (1984); Tsitsiklis et al. (1986) to more recent work that relates to the setting that we consider, which includes Johansson et al. (2007); Nedic and Ozdaglar (2009); Nedić et al. (2009); Johansson et al. (2009); Ram et al. (2010); Lobel and Ozdaglar (2011); Matei and Baras (2011); Boyd et al. (2011); Duchi et al. (2012); Shi et al. (2015); Mokhtari and Ribeiro (2016). From a statistical point of view, however, as emphasised in Shamir and Srebro (2014), distributed learning problems have more structure than general consensus problems, due to the possible statistical similarities in the data owned by different agents, for instance. Aside from the client-server (star network) setting where a central aggregator can coordinate the exchange of information with every other node so that divide and conquer protocols are admissible (Lin and Cevher, 2018), the literature on statistical guarantees for distributed methods seems to have focused exclusively on the investigation of models with explicit regularisation, i.e., when constraints and/or penalty terms are added to the minimisation of the empirical loss function (Agarwal and Duchi, 2011; Zhang et al., 2012; Shamir et al., 2014; Zhang and Lin, 2015; Bijral et al., 2017). The presence of explicit regularisation typically increases the complexity of both the algorithms and the resulting theoretical analysis, particularly for the distributed setting (Lian et al., 2017). For example, constraints can require the need for projection steps which are potentially costly for low-powered sensors, and deriving error bounds that depend on the graph topology for distributed algorithms in the presence of constraints is known to be challenging (Duchi et al., 2012). We are not aware of any result that investigates the performance of distributed and decentralised algorithms (i.e., not divide and conquer methods) on the Test Risk in the absence of explicit regularisation. This is in sharp contrast with the single-machine setting, where recent progress has been made giving optimal statistical learning guarantees for algorithms based on unregularised empirical risk minimisation via notions of implicit regularisation, i.e., proper tuning of algorithmic parameters (Ying and Pontil, 2008; Tarres and Yao, 2014; Dieuleveut and Bach, 2016; Lin et al., 2016a; Lin and Rosasco, 2017).

1.1. Contributions

This paper investigates the learning capabilities of a simple synchronised distributed first-order method for multi-agent learning using notions of implicit regularisation that depend on the topology of the underlying communication graph. We consider the unconstrained and unpenalised empirical risk minimisation problem in the setting where n agents have access to m independent data points coming from the same unknown distribution, and where agents can only exchange information with their neighbours. We consider a synchronised distributed version of stochastic subgradient descent

(Distributed SGD), which is a stochastic variant of one of the most widely-studied first-order method in multi-agent optimisation (Nedic and Ozdaglar, 2009). In the implementation that we look at, at every iteration each agent first performs a standard SGD step, where only one data point is uniformly sampled with replacement among those individually-owned to compute the local subgradient, and then performs a classical synchronised consensus step, where a local exchange of information is implemented via an average of the updated iterates across neighbouring agents. We treat Distributed SGD as a *statistical* device, and look at its performance on unseen data by bounding the Test Error, i.e., the expected value of the excess risk defined as the difference between the Test Risk evaluated at the output of the algorithm and the minimal Test Risk. Under different assumptions on the convex loss function (we consider the standard assumptions of Lipschitz and smoothness) we establish upper bounds for the Test Error of Distributed SGD that exhibit explicit dependence on both the algorithmic tuning parameters (the learning rate and the time horizon) and the graph topology (the spectral gap of the communication matrix). Minimising these upper bounds yields implicit regularisation strategies, allowing to recover the single-machine serial statistical rates by proper tuning of the learning rates and of the time horizon (a.k.a. early stopping) as a function of the network topology. In the case of convex, Lipschitz, and smooth losses, we recover, up to logarithmic terms, the optimal rate of $O(1/\sqrt{nm})$ for single-pass constrained single machine serial SGD (Lan, 2012; Xiao, 2010). In the case of convex and Lipschitz losses, we recover, up to logarithmic terms, the best-known rate of $O(1/(nm)^{1/3})$ for single-machine serial SGD with implicit regularisation Lin et al. (2016a,b).[1] We present numerical experiments to show that the qualitative nature of the upper bounds we derive can be representative of real behaviours.

To establish learning rates for Distributed SGD, we follow the general framework pioneered in the single-machine setting by Bousquet and Bottou (2008) and, in particular, by Hardt et al. (2016). We consider, in the distributed setting, a decomposition of the Test Error which involves the Generalisation Error (i.e., the expected value of the difference between the loss incurred on the training data versus the loss incurred on a new data point) and the Optimisation Error (i.e., the expected value of the error on the training set). To bound the Generalisation Error, we use algorithmic stability or sensitivity as initially put forward by Bousquet and Elisseeff (2002) and later applied for single-machine serial stochastic subgradient descent in Hardt et al. (2016). The notion of stability that we use measures how much the output of an algorithm differs when a single observation is resampled. In our case, as the observations are spread throughout the communication graph, we need to consider stability not only with respect to time (i.e., the iteration time of the algorithm), but also with respect to space (i.e., the communication graph). This technology allows us to establish generalisation bounds for Distributed SGD that do not depend on the topology of the communication graph, and we recover the same type of results that hold in the single machine serial setting. This is in contrast to optimisation bounds for distributed subgradient methods, which typically depend on the graph topology, as initially seen in the work of Johansson et al. (2009, 2007); Duchi et al. (2012). To bound the Optimisation Error, we follow the approach pioneered in Nedic and Ozdaglar (2009) and compare the behaviour of Distributed SGD with its network average, and we take inspiration from the analysis of the network term in the work of Duchi et al. (2012) (in the case of dual methods for constrained problems with Lipschitz losses) to derive upper bounds that depend on the graph topology via the inverse of the spectral gap of the communication matrix. In our setting, as we investigate implicit regularisation strategies, we deal with unconstrained problems

1. Lin et al. (2016b) considers implicit regularisation for gradient descent, although they remark that the analysis can be modified to account for stochastic gradients.

and the evolution of the network-averaged process admits a simple form that facilitates the analysis. This approach avoids the difficulties with the nonlinearity of projection that have been previously challenging in distributed learning models, and that motivated the investigation of dual methods such as in Duchi et al. (2012). The bounds that we establish for the Optimisation Error of Distributed SGD seem novel and are of independent interest.

Finally, our results show that one can also think of the graph itself as a regularisation parameter. To give an example, agents can achieve the same statistical guarantees by trading off communication against iterations: they can choose to communicate by using a low-energy sparse communication protocol per iteration (for instance, communicating using a grid-like protocol even if the underlying topology is that of a complete graph and all agents are connected with each others), but would need to communicate for a longer time horizon in order to be guaranteed to reach the same level of statistical accuracy.

The main contributions of this work are here summarised.

1. **Graph-dependent implicit regularisation.** We propose graph-dependent implicit regularisation strategies for problems in distributed machine learning, specifically, step size tuning and early stopping as a function of the spectral gap of the communication matrix. Our results also show that the graph itself can be interpreted as a regularisation parameter.

2. **Optimal statistical rates using a simple algorithm.** Using implicit regularisation, we show how a simple, primal, unconstrained, first-order method (Distributed SGD) recovers, up to logarithmic terms, centralised statistical rates, in particular matching the optimal rates in the case of smooth loss functions for constrained single-pass serial SGD.

3. To establish statistical rates and control the Test Error of Distributed SGD, we use a distributed version of the error decomposition proposed in Hardt et al. (2016). We establish error bounds on the Generalisation Error and Optimisation Error, respectively.

   (a) **Distributed generalisation bounds.** We establish graph-independent Generalisation Error bounds for Distributed SGD that match those within Hardt et al. (2016) for the single-machine serial case. In the case of convex losses that are Lipschitz and smooth, we prove upper bounds that grow linearly with the number of iterations and step size.

   (b) **Distributed optimisation bounds.** We establish graph-dependent Optimisation Error bounds for Distributed SGD. In the case of convex and Lipschitz loss functions, our analysis is inspired by Nedic and Ozdaglar (2009); Duchi et al. (2012). When smoothness is considered, our analysis is inspired by Bubeck et al. (2015); Dekel et al. (2012).

The remainder of the work is laid out as follows. Section 2 introduces the framework of multi-agent learning. Section 3 introduces the Distributed SGD algorithm. Section 4 presents the main results of this work, Test Error bounds for Distributed SGD with convex, Lipschitz, and either smooth or non-smooth losses. Section 5 presents the specific Generalisation and Optimisation Error bounds, as well as the notion of stability that we use. Section 6 gives a simulation study for the case of smooth losses. Section 7 contains the conclusion. Appendix A provides proofs for all Generalisation and Test Error bounds. Appendix B gives proofs for Optimisation Error bounds under a general first-order stochastic oracle model.

2. Multi-Agent Learning

In this section we introduce the framework of distributed and decentralised machine learning that we consider. We address the case in which agents or nodes in a network are given their own independent data sets and they want to cooperate, by iteratively exchanging information with their neighbours, to develop a good learning model for new unseen data.

Let (V, E) be a simple undirected graph with n nodes, $V = \{1, \dots, n\} \equiv [n]$ being the vertex set and $E \subseteq V \times V$ being the edge set. Let $\mathcal{Z}$ be the space of observations, and to each $v \in V$ let $\mathcal{D}_v := \{Z_{v,1}, \dots, Z_{v,m}\} \in \mathcal{Z}^m$ denote the training set associated to node v, which consists of m i.i.d. data points sampled from a certain unknown distribution supported on $\mathcal{Z}$. Let $\mathcal{D} := \cup_{v \in V} \mathcal{D}_v$ denote the collection of all data points, that is, the entire/global training data set. Let $d > 0$ be a given positive integer, and define $\mathcal{X} = \mathbb{R}^d$. Each agent wants to find a model $x^\star \in \mathcal{X}$ that minimises of the Test Risk r, which is defined as

$$r(x) := \mathbf{E}\, \ell(x, Z).$$

Here, the function $\ell : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ is a given loss function, and $\ell(x, Z)$ represents the loss of the model x on the random sample Z, which represents a new (unseen, independent) data point from the same distribution. We assume that the minimum of r can be achieved. As the distribution of the data is unknown, the expected risk r can not be computed, and a popular approach in machine learning is to consider the empirical risk as a proxy. In the distributed setting, the global empirical risk R is defined as

$$R(x) := \frac{1}{nm} \sum_{v \in V} \sum_{i=1}^{m} \ell(x, Z_{v,i}) = \frac{1}{n} \sum_{v \in V} R_v(x).$$

Here, we have further defined the local empirical risk $R_v(x) := \frac{1}{m} \sum_{i=1}^{m} \ell(x, Z_{v,i})$, for any $v \in V$. Let us denote by $X^\star \in \operatorname{argmin}_{x \in \mathcal{X}} R(x)$ a minimiser of the global empirical risk. In the decentralised setting that we consider, each agent $v \in V$ iteratively exchanges information with their neighbours for a certain amount of time steps t to construct a model $X_v^t \in \mathcal{X}$ that can be a good proxy for the minimiser of the expected risk, i.e., for $x^\star \in \operatorname{argmin}_{x \in \mathcal{X}} r(x)$. A way to assess the quality of a model X_v^t is to consider the Test Error, which we define as the expected value of the excess risk $r(X_v^t) - r(x^\star)$, namely,

$$\mathbf{E}\, r(X_v^t) - r(x^\star).$$

In the next section we introduce the specific distributed algorithm that we consider to generate the models' estimates X_v^t's, and we then present the main results on the bounds for the Test Error. The general paradigm that we adopt to bound the Test Error is given by a generalisation to the distributed setting of the error decomposition given in Hardt et al. (2016) for the single-machine setting. This decomposition allows to bound the Test Error of a model into the sum of two errors: the *Generalisation Error*, which controls the difference between the performance of the model on a new data point and the performance of the model on the training data in $\mathcal{D}$; and the *Optimisation Error*, which controls how well the model optimises the empirical risk.

Proposition 1 (Hardt et al. (2016)) *For each $v \in V$, $t \geq 1$ we have*

$$\underbrace{\mathbf{E}\, r(X_v^t) - r(x^\star)}_{\textit{Test Error}} \leq \underbrace{\mathbf{E}[r(X_v^t) - R(X_v^t)]}_{\textit{Generalisation Error}} + \underbrace{\mathbf{E}[R(X_v^t) - R(X^\star)]}_{\textit{Optimisation Error}}.$$

Proof For completeness, the proof from Hardt et al. (2016) is given in Appendix A.1. ■

By using the error decomposition in Proposition 1, we are able to consider the unregularised empirical risk minimisation problem introduced above and develop implicit regularisation strategies for a simple iterative algorithm, which we introduce next.

Remark 2 (Statistical optimisation) *From the statistical point of view, the distributed setting where each agent is given a subset of the data has received a lot of attention in the literature (see introduction), though most of the literature on statistical optimisation has focused on the client-server (also known as master-slave) architecture typical of data centers, where a central aggregator in the network (the server) can communicate with every other nodes (the clients) and can thus coordinate the processing and exchange of information. This amounts to a star network topology that can be used to model shared-memory protocols. This type of architecture makes divide-and-conquer strategies possible, and most of the literature on statistical optimisation has focused on investigating statistical rates on the Test Error for one-shot-averaging techniques. In this work, we focus on the decentralised setting where all nodes iteratively perform the same type of computations and communications with respect to the underlying graph structure, without the presence of any special node. We are not aware of any prior work that directly investigates the statistical performance of decentralised methods on the Test Error. Most of the literature on decentralised methods seem to have focused on bounding the Optimisation Error on the training data, as we explain in Remark 3.*

Remark 3 (Consensus optimisation) *From the optimisation point of view, the literature on multi-agent learning has largely focused on the investigation of the Optimisation Error via consensus methods in the presence of explicit regularisation, typically in the form of a convex constraint set $\mathcal{R}$ (see literature review in the introduction). Statistically, this approach is justified, for instance, by the distributed version of the classical error decomposition given in Bousquet and Bottou (2008):*

$$\underbrace{\mathbf{E}\, r(X_v^t) - r(x^\star)}_{\textit{Test Error}} \leq 2 \underbrace{\mathbf{E} \sup_{x\in\mathcal{R}} |r(x) - R(x)|}_{\textit{Uniform Generalisation Error}} + \underbrace{\mathbf{E}[R(X_v^t) - R(X_{\mathcal{R}}^\star)]}_{\textit{Regularised Optimisation Error}} + \underbrace{r(x_{\mathcal{R}}^\star) - r(x^\star)}_{\textit{Approximation Error}},$$

with $x_{\mathcal{R}}^\star \in \operatorname{argmin}_{x\in\mathcal{R}} r(x)$ and $X_{\mathcal{R}}^\star \in \operatorname{argmin}_{x\in\mathcal{R}} R(x)$. In this setting, consensus optimisation deals with algorithms that minimise the quantity $R(X_v^t) - R(X_{\mathcal{R}}^\star)$, where $R(x) = \frac{1}{n}\sum_{v\in V} R_v(x)$. Bounds on the Regularised Optimisation Error can then be combined with bounds on the Uniform Generalisation Error using notions of complexity for the constraint set $\mathcal{R}$ (e.g., VC dimension, Rademacher complexity, etc.). As highlighted in Shamir and Srebro (2014), and as we mentioned in the introduction, however, distributed learning problems have more structure than general consensus problems, as the local functions R_v are random and have a specific design. In this work, we analyse a stochastic algorithm that is tailor-made for distributed learning problems (not for general consensus problems), and use the error decomposition in Proposition 1 to develop implicit regularisation strategies for the unregularised empirical risk minimisation problem.

3. Distributed Stochastic Subgradient Descent

The algorithm that we consider to generate the model estimates X_v^t's assumes that each node $v \in V$ can query subgradients $\partial\ell$ of the loss function ℓ with respect to the first parameter, evaluated at

points in the local data set $\mathcal{D}_v$. We consider the stochastic setting where at each time step agent v does not evaluate the full subgradient of the local empirical risk R_v, but instead only a subgradient $\partial\ell$ at a single randomly chosen sample in the locally-owned data set $\mathcal{D}_v$. This is well tailored to situations where m is large, as this reduces the per-iteration complexity to a constant factor.

The algorithm is defined as follows. Let $\partial\ell(x, Z_{v,k})$ represent an element of the subgradient of $\ell(\,\cdot\,, Z_{v,k})$ at x, with $k \in \{1, \dots, m\} \equiv [m]$. Let $P \in \mathbb{R}^{n\times n}$ be a doubly stochastic matrix supported on the graph (V, E), that is, $P_{ij} \neq 0$ only if $\{i, j\} \in E$. Distributed stochastic subgradient descent (Distributed SGD) generates a collections of vectors $\{X_v^s\}_{v\in V, s\geq 1}$ in $\mathcal{X}$ as follows. Given initial vectors $\{X_v^1\}_{v\in V}$, possibly random, for $s \geq 1$,

$$X_v^{s+1} = \sum_{w\in V} P_{vw}(X_w^s - \eta\partial\ell(X_w^s, Z_{w,K_w^{s+1}})), \tag{1}$$

where for each $v \in V$, $\{K_v^2, K_v^3, \dots\}$ is a collection of i.i.d. random variables uniform in $[m]$, and $\eta > 0$ is the step size. The above algorithm can be described as performing two steps: a stochastic gradient update $Y_w^{s+1} = X_w^s - \eta\partial\ell(X_w^s, Z_{w,K_w^{s+1}})$, and a synchronised consensus step $\sum_{w\in V} P_{vw}Y_w^{s+1}$. This framework for decentralised optimisation (albeit for a slightly different protocol, see remark 4) has been largely explored with the early works of Nedic and Ozdaglar (2009); Ram et al. (2009); Lobel and Ozdaglar (2011); Duchi et al. (2012). The fact that we consider implicit regularisation strategies allows us to focus on the unconstrained risk minimisation problem. In turn, this allows us to consider an algorithm that is much simpler to analyse than the ones previously considered in the literature, avoiding projections or dual approaches (see introduction for the relevant literature review). We also highlight the randomised sampling mechanism in algorithm (1), which is tailor-made for the machine learning problem at hand and not for generic consensus problems.

Remark 4 *In the stochastic setting, the protocol put forward by Nedic and Ozdaglar (2009) updates the iterates as* $X_v^{s+1} = \sum_{w\in V} P_{vw}X_w^s - \eta\partial\ell(X_v^s, Z_{v,K_v^{s+1}})$*, which is slightly different from the protocol that we consider where also the gradients are averaged across neighbours. The two main motivations for the original protocol are that it is fully decentralised, in that nodes are only required to communicate locally, and that it reduces to a consensus protocol to solve network averaging problems when* $\ell = 0$*. The protocol* (1) *that we consider preserves these properties and it makes the error analyses simpler. The difference between these two protocols in a general setting has been investigated in the literature, see Sayed (2014) for instance.*

In the next section we present results on the performance of Distributed SGD under various assumptions on the loss function ℓ.

4. Results

This section presents the main results of this work: Test Error bounds for Distributed SGD with smooth and non-smooth losses, Section 4.1 and Section 4.2, respectively.

Henceforth, let $\|\cdot\|$ be the ℓ_2 norm. A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be L-Lipschitz, with $L > 0$, if $|f(x) - f(y)| \leq L\|x - y\|$ for all $x, y \in \mathbb{R}^d$, and β-smooth, with $\beta > 0$, if $\|\nabla f(x) - \nabla f(y)\| \leq \beta\|x - y\|$ for all $x, y \in \mathbb{R}^d$. Let $\sigma_2(P)$ be the second largest eigenvalue in absolute value for the matrix P. Unless stated otherwise, we use the big-O notation $O(\,\cdot\,)$ to

denote order of magnitudes up to constants in n and m, and the notation $\widetilde{O}(\,\cdot\,)$ to denote order of magnitudes up to both constants and logarithmic terms in n and m. Equality modulo constants and logarithmic terms is denoted by $\simeq$.

4.1. Smooth Losses

We analyse the statistical rates for smooth losses. First, we present the Test Error bound in its full form. Then, we present a corollary that summarises the order of magnitudes of the bounds obtained under different choices of implicit regularisation, tuning the step size and the stopping time as a function of the graph topology. Full details are given in Appendix A.

For smooth losses, we present a bound that depends on both the variance of the gradient estimates and the statistical deviations between the local empirical losses $\{R_v\}_{v\in V}$. Let $\sigma, \kappa > 0$ be such that the following holds for any $v \in V$ and $s \geq 1$,

$$\mathbf{E}\big[\|\nabla \ell(X_v^s, Z_{v,K_v^{s+1}}) - \nabla R_v(X_v^s)\|^2\big] \leq \sigma^2, \tag{2}$$

$$\mathbf{E}\big[\|\nabla \ell(X_v^s, Z_{v,K_v^{s+1}}) - \frac{1}{n}\sum_{w\in V} \nabla R_w(X_w^s)\|^2\big] \leq \kappa^2. \tag{3}$$

The quantity σ^2 in (2) yields a uniform control on the variance of the stochastic gradients, while the quantity κ^2 in (3) yields a uniform control on both the variance of the gradients and the deviation between local objectives. Note that if $\ell(\,\cdot\,, z)$ is L-Lipschitz for any $z \in \mathcal{Z}$, then both σ^2 and κ^2 are bounded by $4L^2$ by the triangle inequality. A detailed discussion of these assumptions is given in Appendix B in the more general context of stochastic optimisation.

Theorem 5 (Test Error bounds for convex, Lipschitz, and smooth losses) *Assume that for any $z \in \mathcal{Z}$ the function $\ell(\,\cdot\,, z)$ is convex, L-Lipschitz, β-smooth and satisfies* (2) *and* (3)*. Let $X_v^1 = 0$ for all $v \in V$, $\|X^\star\| \leq G$. Then, Distributed SGD with $\eta = 1/(\beta + 1/\rho)$, $\rho > 0$, and $\eta\beta \leq 2$, yields, for any $v \in V$ and $t \geq 1$,*

$$\mathbf{E}\, r\Big(\frac{1}{t}\sum_{s=1}^{t} X_v^{s+1}\Big) - r(x^\star) \leq \underbrace{\frac{L^2}{nm(\beta + 1/\rho)}(t+1)}_{\textit{Generalisation Error bound}}$$
$$+ \underbrace{\frac{\rho}{2}\sigma^2 + \frac{(\beta+1/\rho)G^2}{2t} + \frac{3\kappa}{\beta+1/\rho}\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)}\Big(L + \frac{3}{2}\frac{\beta(3+\beta\rho)\kappa}{\beta+1/\rho}\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)}\Big)}_{\textit{Optimisation Error bound}}.$$

Proof See Appendix A.5. ■

We highlight that the Generalisation Error bound is independent of the graph topology, while the Optimisation Error bound naturally depends upon inverse of the spectral gap of the communication matrix: $1/(1-\sigma_2(P))$. The following corollary gives the order of magnitudes for the Test Error bounds obtained with three different choices of step size and corresponding early stopping. The different choices for the parameter $\rho > 0$ correspond to the following (modulo the simplifications used to perform the minimisations, as explained in detail in Section A.6):

- $\rho^\star$ is the choice for serial SGD, see for instance Dekel et al. (2012); Bubeck et al. (2015);

- $\rho^\star_{\text{Opt}}$ is the choice that minimises the Optimisation Error bound in Theorem 5;
- $\rho^\star_{\text{Test}}$ is the choice that minimises the Test Error bound in Theorem 5.

Corollary 6 (Implicit regularisation for convex, Lipschitz, and smooth losses) *In the setting of Theorem 5, the following holds for different choices of ρ, function of the time horizon t:*

ρ	Size	Test Error at ρ, t	Test Error at $\rho, t^\star(\rho)$
$\rho^\star$	$O\Big(\frac{1}{\sqrt{t}}\Big)$	$\widetilde{O}\Big(\frac{1}{(1-\sigma_2(P))\sqrt{t}} + \frac{\sqrt{t}}{nm}\Big)$	$\widetilde{O}\Big(\frac{1}{\sqrt{nm(1-\sigma_2(P))}}\Big)$
$\rho^\star_{\text{Opt}}$	$\widetilde{O}\Big(\sqrt{\frac{1-\sigma_2(P)}{t}}\Big)$	$\widetilde{O}\Big(\frac{1}{\sqrt{t(1-\sigma_2(P))}} + \frac{\sqrt{t(1-\sigma_2(P))}}{nm}\Big)$	$\widetilde{O}\Big(\frac{1}{\sqrt{nm}}\Big)$
$\rho^\star_{\text{Test}}$	$\widetilde{O}\Big(\frac{1}{\sqrt{t}}\frac{1}{\sqrt{\frac{1}{1-\sigma_2(P)}+\frac{t}{nm}}}\Big)$	$\widetilde{O}\Big(\frac{1}{\sqrt{t(1-\sigma_2(P))}} + \frac{1}{\sqrt{nm}}\Big)$	$\widetilde{O}\Big(\frac{1}{\sqrt{nm}}\Big)$

Table 1: $t^\star(\rho^\star) \simeq t^\star(\rho^\star_{\text{Opt}}) \simeq t^\star(\rho^\star_{\text{Test}}) \simeq nm/(1-\sigma_2(P))$.

Proof See Appendix A.6. ■

We note that the Test Error bound given by the choice $\rho^\star_{\text{Test}}$ is the only one that is guaranteed to converge as the number of iterations t goes to infinity. With this choice, $t^\star(\rho^\star_{\text{Test}}) \simeq nm/(1-\sigma_2(P))$ iterations are guaranteed to reach the rate $\widetilde{O}(1/\sqrt{nm})$. Minimising (approximately) with respect to time the Test Error bounds that are obtained with the choices $\rho^\star$ and $\rho^\star_{\text{Opt}}$ gives early stopping rules with the same order of iterations, i.e., $t^\star(\rho^\star) \simeq t^\star(\rho^\star_{\text{Opt}}) \simeq nm/(1-\sigma_2(P))$. The choices $\rho^\star_{\text{Test}}$ and $\rho^\star_{\text{Opt}}$ with early stopping yield, up to logarithmic terms, the optimal rate $O(1/\sqrt{nm})$ for single-pass constrained serial SGD (Lan, 2012; Xiao, 2010). On the other hand, the choice $\rho^\star$ that aligns with serial SGD, with no dependence on the graph topology, yields a suboptimal statistical guarantee with a rate $\widetilde{O}(1/\sqrt{nm(1-\sigma_2(P))})$.

Remark 7 (Knowledge of Network Spectrum) *Algorithmic parameter choices in Table 1 depend on the network through the spectral gap of the communication matrix $1-\sigma_2(P)$. While outside the scope of this work, this quantity can be estimated in a decentralised manner, see for instance (Yang et al., 2010; Yang and Tang, 2015) and references therein.*

Remark 8 (Early Stopping with a Constant Step Size) *When performing early stopping a step size constant in the number of iterations is commonly chosen so a single instance of single-machine serial SGD is required. Theorem 5 demonstrates optimal statistical rates up to logarithmic factors can be achieved for Distributed SGD when choosing the step size $\rho = O((1-\sigma_2(P)))/\sqrt{nm})$ and iterations $t = O(nm/(1-\sigma_2(P)))$. For the calculation of this fact see Appendix A.8.*

4.2. Non-Smooth Losses

We now analyse the statistical rates for non-smooth losses. Before presenting the results, we introduce and motivate the technical assumptions that we need.

Assumptions 1

(a) There exist constants $C \leq B$ such that for any $z \in \mathcal{Z}$ the loss function $\ell(\,\cdot\,, z)$ is bounded from above at zero, i.e., $\ell(0, z) \leq B$, and is uniformly bounded from below, i.e., $C \leq \ell(x, z)$ for any $x \in \mathbb{R}^d$;

(b) There exists a constant $D \geq 0$ such that for any $z_1, \ldots, z_{nm} \in \mathcal{Z}$ and any $\widetilde{\mathcal{X}} \subseteq \mathcal{X}$ we have

$$\mathbf{E} \sup_{x \in \widetilde{\mathcal{X}}} \frac{1}{nm} \sum_{i=1}^{nm} \sigma_i \ell(x, z_i) \leq D \frac{\sup_{x \in \widetilde{\mathcal{X}}} \|x\|}{\sqrt{nm}},$$

where $\{\sigma_i\}_{i \in [nm]}$ is a collection of independent Rademacher random variables, namely, $\mathbf{P}(\sigma_i = 1) = \mathbf{P}(\sigma_i = -1) = 1/2$.

Assumption *(a)* is a common boundedness assumption for controlling the norm of the iterates of gradient descent algorithms through a centring argument. Assumption *(b)* represents a control on the Rademacher complexity of the function class $\{\ell(x, \,\cdot\,) : x \in \mathcal{X}\}$ with respect to the ℓ_2 norm. These assumptions are satisfied, for instance, in the setting of supervised learning with linear predictors, bounded data, and hinge loss (with is convex, Lipschitz, and non-smooth). See Remark 11 below.

First, we present the Test Error bound for non-smooth losses under Assumptions 1. Then, we present a corollary that summarises the order of magnitudes of the bounds obtained under different choices of implicit regularisation, tuning the step size and the stopping time as a function of the graph topology. Full details are given in Appendix A.

Theorem 9 (Test Error bounds for convex and Lipschitz losses) *Assume that for any $z \in \mathcal{Z}$ the loss function $\ell(\,\cdot\,, z)$ is convex and L-Lipschitz. Consider Assumptions 1. Let $X_v^1 = 0$ for all $v \in V$, $\|X^\star\| \leq G$. Then, Distributed SGD with $\eta > 0$ yields, for any $v \in V$ and $t \geq 1$,*

$$\mathbf{E}\, r\Big(\frac{1}{t}\sum_{s=1}^{t} X_v^s\Big) - r(x^\star) \leq \underbrace{2D\sqrt{\frac{(t-1)(\eta^2 L^2 + 2\eta(B-C))}{nm}}}_{\textit{Generalisation Error bound}} + \underbrace{\frac{19}{2}\frac{\eta L^2 \log(t\sqrt{n})}{1-\sigma_2(P)} + \frac{G^2}{2\eta t}}_{\textit{Optimisation Error bound}}.$$

Proof See Appendix A.5. ■

The following corollary gives the order of magnitudes for the Test Error bound obtained with three different choices of step size and corresponding early stopping. The different choices for the step size $\eta > 0$ correspond to the following (modulo the simplifications used to perform the minimisations, as explained in detail in Section A.7):

- $\eta^\star$ is the choice for serial SGD, see for instance Bubeck et al. (2015);
- $\eta^\star_{\mathrm{Opt}}$ is the choice that minimises the Optimisation Error bound in Theorem 9;
- $\eta^\star_{\mathrm{Test}}$ is the choice that minimises the Test Error bound in Theorem 9.

Corollary 10 (Implicit regularisation for convex and Lipschitz losses) *In the setting of Theorem 9, the following holds for different choices of η, function of the time horizon t:*

η	Size	Test Error at η, t	Test Error at $\eta, t^\star(\eta)$
$\eta^\star$	$O\Big(\frac{1}{\sqrt{t}}\Big)$	$\widetilde{O}\Big(\frac{1}{(1-\sigma_2(P))\sqrt{t}} + \sqrt{\frac{\sqrt{t}}{nm}}\Big)$	$\widetilde{O}\Big(\frac{1}{(nm(1-\sigma_2(P)))^{1/3}}\Big)$
$\eta^\star_{\text{Opt}}$	$\widetilde{O}\Big(\sqrt{\frac{1-\sigma_2(P)}{t}}\Big)$	$\widetilde{O}\Big(\frac{1}{\sqrt{t(1-\sigma_2(P))}} + \sqrt{\frac{\sqrt{t(1-\sigma_2(P))}}{nm}}\Big)$	$\widetilde{O}\Big(\frac{1}{(nm)^{1/3}}\Big)$
$\eta^\star_{\text{Test}}$	$\widetilde{O}\Big(\frac{1}{\sqrt{t}}\frac{1}{\sqrt{\frac{1}{1-\sigma_2(P)}+\frac{t}{(nm)^{2/3}}}}\Big)$	$\widetilde{O}\Big(\frac{1}{\sqrt{t(1-\sigma_2(P))}} + \frac{1}{(nm)^{1/3}}\Big)$	$\widetilde{O}\Big(\frac{1}{(nm)^{1/3}}\Big)$

Table 2: $t^\star(\eta^\star) \simeq (nm)^{2/3}/(1-\sigma_2(P))^{4/3}$ and $t^\star(\eta^\star_{\text{Opt}}) \simeq t^\star(\eta^\star_{\text{Test}}) \simeq (nm)^{2/3}/(1-\sigma_2(P))$.

Proof See Appendix A.7. ■

Corollary 10 shows asymptotic behaviours for the Test Error bounds (as a function of time t upon different choices of the step size) that are analogous to the ones established in Corollary 6 in the case of smooth losses. In particular, as in Corollary 6, the step sizes accounting for the graph topology, i.e., $\eta^\star_{\text{Test}}$ and $\eta^\star_{\text{Opt}}$, give improved statistical rates over the step size independent of the graph topology $\eta^\star$.

The statistical rate obtained by both $\eta^\star_{\text{Test}}$ and $\eta^\star_{\text{Opt}}$, upon performing early stopping, matches, up to logarithmic terms, the best-known rate of $O(1/(nm)^{1/3})$ obtained by serial SGD with implicit regularisation (Lin et al., 2016a). Differing from the smooth case, additional iterations with respect to the graph topology are required for the step size independent of the graph topology $\eta^\star$ to achieve its best statistical rates (as prescribed by our upper bounds), when compared to step sizes accounting for the topology $\eta^\star_{\text{Test}}$ and $\eta^\star_{\text{Opt}}$. As highlighted in (Lin et al., 2016a), we note that these rates are not sharp, leaving it to future work to obtain better bounds.

Remark 11 *Assumptions 1 is satisfied in the setting of supervised learning with bounded data, linear predictors, and hinge loss, for instance. In this setting, each observation $z \in \mathcal{Z}$ decomposes into a d-dimensional feature vector and a real-valued response, i.e., $z = \{w, y\}$ with $w \in \mathcal{W} \subset \mathbb{R}^d$ and $y \in \mathcal{Y} \subset \mathbb{R}$ such that $\|w\| \leq D_{\mathcal{W}} < \infty$, and $|y| \leq D_{\mathcal{Y}} < \infty$. The linear predictors are parametrised by $x \in \widetilde{\mathcal{X}} \subseteq \mathcal{X} = \mathbb{R}^d$, i.e., $w \to w^\top x$, and the loss function is of the form $\ell(x, z) = \tilde{\ell}(w^\top x, y)$ with the function $\tilde{\ell} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ measuring the discrepancy between the predicted response $w^\top x$ and the observed response y. For the hinge loss, $\tilde{\ell}(\tilde{y}, y) = \max(0, 1 - \tilde{y}y)$. Assumption 1 (a) is satisfied with $B = 1$ and $C = 0$. By Talagrand's contraction lemma and standard results on the Rademacher complexity of linear predictors, assumption (b) is satisfied with $D = D_{\mathcal{Y}} D_{\mathcal{W}}$, as the hinge loss $\tilde{\ell}(\,\cdot\,, y)$ is $|y|$-Lipschitz. Also the Lipschitz constant in Theorem 9 reads $L = D$, as $|\ell(x_1, z) - \ell(x_2, z)| \leq D_{\mathcal{Y}} |(x_1 - x_2)^\top w| \leq D_{\mathcal{Y}} D_{\mathcal{W}} \|x_1 - x_2\|$ by the Cauchy-Schwarz's inequality.*

5. Generalisation and Optimisation Error Bounds

In this section we present the Generalisation and Optimisation Error bounds that yield the Test Error bounds presented within Section 4. Section 5.1 begins with the stability analysis used to derive the Generalisation Error bounds for smooth losses. This is followed by the Generalisation Error bound for non-smooth losses in Section 5.2. Finally, Section 5.3 presents Optimisation Error bounds for both classes of losses.

5.1. Generalisation Error Bound for Smooth Losses through Stability

To bound the Generalisation Error for smooth losses we utilise its link with stability. This has previously been investigated in Rogers and Wagner (1978); Kearns and Ron (1999); Bousquet and Elisseeff (2002); Mukherjee et al. (2006); Shalev-Shwartz et al. (2010), with Bousquet and Elisseeff (2002) and Hardt et al. (2016) providing the work upon which we rely. Specifically, Hardt et al. (2016) investigated the Generalisation Error of serial SGD in the multi-pass setting, giving, in the case of convex, Lipschitz, and smooth losses, upper bounds that grow linearly with the number of iterations and step size. The method used is algorithmic stability (or sensitivity) as introduced in Bousquet and Elisseeff (2002). This method investigates the deviation of an algorithm when a single data point in the data set $\mathcal{D}$ is resampled. By iterating through all of the observations, accounting for the deviation in each case, the Generalisation Error is then equal to the average deviation, as we see next. In our case the observations are spread throughout a graph, so the deviations of the algorithm depends on the location of the observation that is resampled.

For each $w \in V$ and $k \in [m]$, let $\widetilde{Z}_{w,k}$ be a resampled (independent) observation coming from the same data distribution. Let $\widetilde{X}(w,k)_v^t$ denote the output of Distributed SGD at node v after t iterations when the algorithm is run on the perturbed data set $\{\mathcal{D} \backslash Z_{w,k}\} \cup \widetilde{Z}_{w,k}$ in which the k-th observation for node w, i.e., $Z_{w,k}$, is replaced by $\widetilde{Z}_{w,k}$. The Generalisation Error is then equal to the average mean deviance of the loss function evaluated at the perturbed outputs.

Proposition 12 *For any $v \in V$ and $t \geq 1$,*

$$\mathbf{E}[r(X_v^t) - R(X_v^t)] = \frac{1}{nm} \sum_{w \in V} \sum_{k=1}^{m} \mathbf{E}[\ell(X_v^t, \widetilde{Z}_{w,k}) - \ell(\widetilde{X}(w,k)_v^t, \widetilde{Z}_{w,k})].$$

Proof The proof is given in Appendix A.2. ■

The identity in Proposition 12 involves a double sum over the mean deviations of the algorithm applied to locally perturbed data sets: one sum relates to the graph location where the perturbation is supported (w), and the other sum relates to the index of the perturbed data point at that location (k). Each *individual* deviation depends on the graph topology via the location of the resampled observation w relative to the node of reference v. This dependence is captured by the bound that we give in Proposition 18 in Appendix A.3.2, where we show that the non-expansive property of the gradient descent update in the smooth case controls the spatial propagation of the deviation across the network via the term $\sum_{s=1}^{t-1}(P^s)_{vw}$. Proposition 12 involves the *average* across all deviations, and once the summation over $w \in V$ is considered, we get a final bound that increases linearly with time but does not depend on the graph topology, as we state next.

Lemma 13 (Generalisation Error bound for convex, Lipschitz, and smooth losses) *Assume that for any $z \in \mathcal{Z}$ the function $\ell(\,\cdot\,, z)$ is convex, L-Lipschitz, and β-smooth. Let $X_v^1 = 0$ for all $v \in V$. Then, Distributed SGD with $\eta\beta \leq 2$ yields, for any $v \in V$ and $t \geq 1$,*

$$\mathbf{E}[r(X_v^t) - R(X_v^t)] \leq \frac{2\eta L^2}{nm}(t-1).$$

Proof See Appendix A.3. ■

For completeness, and to fully establish in the decentralised case the results derived in Hardt et al.

(2016) in the single machine case, we include in Appendix A.3 also the time-uniform Generalisation Error bound for the constrained and strongly-convex case. In this case, the *contraction* property of the gradient descent update controls the spatial propagation of the deviation across the network via the term $\sum_{s=1}^{t-1} \iota^s (P^s)_{vw}$, for a given $\iota < 1$. Once the summation over $w \in V$ in Proposition 12 is taken, we get a final bound that does not depend on time, nor on the graph topology. The bounds that we give are identical to the ones in Hardt et al. (2016) for a single agent with nm observations.

5.2. Generalisation Error Bound for Non-Smooth Losses through Rademacher Complexity

In the case of non-smooth losses we follow the approach used in Lin et al. (2016a) for the single-machine case that involves controlling the Generalisation Error by using standard Rademacher complexity's arguments through Assumption 1 *(b)* and bounding the norm of the iterates $\|X_v^t\|$ as a function of the parameters of the algorithm.

Lemma 14 (Generalisation Error for convex and Lipschitz losses) *Assume that for any $z \in \mathcal{Z}$ the loss function $\ell(\,\cdot\,, z)$ is convex and L-Lipschitz. Consider Assumptions 1. Let $X_v^1 = 0$ for all $v \in V$. Then, Distributed SGD yields, for any $v \in V$ and $t \geq 1$,*

$$\mathbf{E}[r(X_v^t) - R(X_v^t)] \leq 2D\sqrt{\frac{(t-1)(\eta^2 L^2 + 2\eta(B-C))}{nm}}.$$

Proof See Appendix A.4. ■

We now go on to give Optimisation Error bounds which, once combined the Generalisation Error bounds in Section 5.1 and 5.2, give the Test Error bounds presented within Section 4.

5.3. Optimisation Error Bounds

In this section we present Optimisation Error bounds for Distributed SGD with convex, Lipschitz, and either smooth or non-smooth losses. These results follow from theorems proved within Appendix B under the more general setting of the first-order stochastic oracle model. We note that constants within these bounds have not been optimised.

The bounds that we derive are proved using the techniques developed in Nedić et al. (2009) and, in particular, in Duchi et al. (2012), where the evolution of the algorithm X_v^s is compared against the evolution of its network average $\overline{X}^s := \frac{1}{n}\sum_{v\in V} X_v^s$ to derive graph-dependent error bounds. Appendix B contains the full scheme of the proof, along with the error decomposition into a network term, an optimisation term, and a gradient noise term (only in the smooth case). As previously emphasised, the fact that we investigate implicit regularisation strategies allows us to deal with unconstrained problems, and in this case the evolution of the network-averaged process $\overline{X}^s$ admits a simple form that facilitates the analysis. This approach avoids the difficulties with the nonlinearity of projection that have been previously challenging in distributed learning models, and that motivated the investigation of dual methods such as in Duchi et al. (2012).

We start with the case of Lipschitz and smooth losses. The proof for this case is inspired from the proof for serial SGD applied to smooth objectives, specifically, Theorem 6.3 in Bubeck et al. (2015), itself extracted from Dekel et al. (2012). The bound that we present depends upon both the quantity σ and the quantity κ defined, respectively, in (2) and (3).

Lemma 15 (Optimisation Error bound for convex, Lipschitz, and smooth losses) *Assume that for any $z \in \mathcal{Z}$ the function $\ell(\,\cdot\,, z)$ is convex, L-Lipschitz, β-smooth and satisfies* (2) *and* (3)*. Let $X_v^1 = 0$ for all $v \in V$, $\|X^\star\| \le G$. Then, Distributed SGD with $\eta = 1/(\beta + 1/\rho)$ and $\rho > 0$, yields, for any $v \in V$ and $t \ge 1$,*

$$\mathbf{E}\Big[R\Big(\frac{1}{t}\sum_{s=1}^{t} X_v^{s+1}\Big) - R(X^\star)\Big] \le \frac{\rho}{2}\sigma^2 + \frac{(\beta+1/\rho)G^2}{2t} + \frac{3\kappa}{\beta+1/\rho}\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)}\Big(L + \frac{3}{2}\frac{\beta(3+\beta\rho)\kappa}{\beta+1/\rho}\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)}\Big).$$

Proof The result follows from Corollary 27 in Appendix B and from Section B.4. ■

Next is the Optimisation Error bound for non-smooth losses, inspired from Duchi et al. (2012).

Lemma 16 (Optimisation Error bound for convex and Lipschitz losses) *Assume that for any $z \in \mathcal{Z}$ the function $\ell(\cdot, z)$ is convex and L-Lipschitz. Let $X_v^1 = 0$ for all $v \in V$, $\|X^\star\| \le G$. Then, Distributed SGD yields, for any $v \in V$ and $t \ge 1$,*

$$\mathbf{E}\Big[R\Big(\frac{1}{t}\sum_{s=1}^{t} X_v^{s}\Big) - R(X^\star)\Big] \le \frac{\eta L^2}{2}\Big(19\frac{\log(t\sqrt{n})}{1-\sigma_2(P)}\Big) + \frac{G^2}{2\eta t}.$$

Proof The result follows from Corollary 25 in Appendix B and from Section B.4. ■

When optimising either of these bounds with respect to ρ or η, a rate no better than $O(1/\sqrt{t})$ can be obtained, matching the rate of stochastic gradient descent in the single-machine case. From the bound in Lemma 15, however, we note that if $\sigma = \kappa = 0$ then the accelerated rate of $O(1/t)$ can be obtained, matching the rate of full-gradient descent in the single-machine case. For a general discussion on these lines, we refer to Appendix B and to Remark 22 in particular.

6. Numerical Experiments

In this section we provide a simulation study to investigate if the previously proven bounds can be representative of real behaviours. Specifically, we investigate the Test Error bounds given in Corollary 6 for convex, Lipschitz, and smooth losses. We start by introducing the notation and quantities of interest in Section 6.1, then we present the results of the experiments in Section 6.2.

6.1. Setup

As we want to minimise the expected risk $r(x) = \mathbf{E}\,\ell(x, Z)$ but a closed form expression is typically not available, we use a Monte Carlo approximation constructed from an independent out of sample data set $\{Z_j\}_{j\in[\widehat{N}]}$, namely, $\hat{r}(x) := \frac{1}{\widehat{N}}\sum_{j=1}^{\widehat{N}} \ell(x, Z_j)$. Given t iterations of the Distributed SGD algorithm, we denote the ergodic average of the iterates by $\widehat{X}_v^t := \frac{1}{t}\sum_{s=1}^{t} X_v^s$, for $v \in V$. We investigate the Out of Sample Risk defined as $\max_{v\in V} \hat{r}(\widehat{X}_v^t)$, which is set to be a proxy for the Test Risk for Distributed SGD, as defined in Section 2. We recall that the Test Error is defined as the expectation of the Test Risk minus the minimum expected risk $r(x^\star)$, which is a constant. Therefore, modulo a constant shift, Out of Sample Risk is also a proxy for the Test Error.

Given a graph (V, E) with $n = |V|$ nodes, let $A \in \mathbb{R}^{n\times n}$ be its adjacency matrix defined as $A_{vw} := 1$ if $\{v, w\} \in E$ and $A_{vw} := 0$ otherwise. For each $v \in V$, let $d_v = \sum_{w\in V} A_{vw}$ denote the degree of node v, $d_{\max} = \max_{v\in V} d_v$ the maximum degree, and $D = \text{diag}(d_1, \ldots, d_n)$ the diagonal degree matrix. We consider the doubly stochastic matrix $P = I - \frac{1}{d_{\max}+1}(D - A)$. This choice is standard in distributed optimisation (see Shah (2009), for instance). In this case, the spectral gap is known to be of the following orders (see Duchi et al. (2012), for instance):

$$O\left(\frac{1}{\sqrt{1-\sigma_2(P)}}\right) = \begin{cases} n & \text{Cycle} \\ \sqrt{n} & \text{Grid} \\ 1 & \text{Complete Graph} \end{cases}$$

We adopt the following parametrisation: $O(1/\sqrt{1-\sigma_2(P)}) = O(n^\alpha)$, for $\alpha \geq 0$. These topologies are typical of those used in decentralised networks (Shah, 2009; Dimakis et al., 2010).

We consider an instance of logistic regression in supervised learning, where for a given positive integer d, we have $Z = \{W, Y\}$ with the feature vector $W \in \mathbb{R}^d$ and the label $Y \in \{-1, 1\}$, and the parameter of interest is $X \in \mathbb{R}^d$. The loss function in this case is given by

$$\ell(X, Z) = \log(1 + e^{-Y\times\langle X,W\rangle}),$$

where $\langle X, W\rangle = X^\top W = \sum_{i=1}^d X_i W_i$. Given the node count n and m locally-owned data points, a simulated data set with a total of $N = mn$ observations $\{Z_i\}_{i\in[N]}$ are sampled following the experiment within Duchi et al. (2012). Specifically, a true parameter $X^{\star\star}$ is sampled from a standard d-dimensional Gaussian $\mathcal{N}(0, I)$, the feature vectors W_i's are sampled uniformly from the unit sphere $\{w \in \mathbb{R}^d : \|w\| \leq 1\}$, and the responses are set as $Y_i = \text{sign}(\langle W_i, X^{\star\star}\rangle)$ where $\text{sign}(a) = 1$ if $a \geq 0$ and -1 if $a < 0$. The data set is then randomly spread across the graph with each node getting m samples. It can easily be seen that the Lipschitz parameter is $L = 1$ and the smoothness parameter is $\beta = 1/4$. Parameters depending upon the gradient noise were upper bounded by distribution-independent quantities and set to $\sigma^2 \to 4L^2$ and $\kappa \to L$ (see Proposition 23 in Appendix B for the interplay between L and κ as far as bounding the network term is concerned). A solution $X^\star$ to the empirical risk minimisation rule is calculated with tolerance 10^{-15} using the `lbfgs` solver within the `LogisticRegression` function of the python library `scikit` (Pedregosa et al., 2011). We set $G = \|X^\star\|$. Dimension and Monte Carlo estimate size are set to $d = 100$ and $\widehat{N} = 1000$, respectively. We investigate the performance of Distributed SGD in two sample size regimes $m = 2$ and $m = 100$, which we now go on to describe in more detail.

6.2. Experimental Results - Small Sample Regime

This setting explores the small sample size regime where by agent receive $m = 2$ samples each. Distributed SGD is run for 15 different time horizons t, between 10^2 and either 10^7 or $10^{6.5}$ for graph sizes $n = 3^2$ or $n = 10^2$, respectively. All runs are initialised from $X_v^1 = 0$ for all $v \in V$. Comparisons are made for three choices of the step size, as prescribed in Corollary 6, and for three choices of the graph topology: complete graph ($\alpha = 0$), grid ($\alpha = 1/2$), and cycle ($\alpha = 1$). Referring to the *upper* bounds in Corollary 6, we outline what we expect to see plotting the Test Error against the time horizon t, with $\log - \log$ scales, across the three different step sizes:

- $\rho^\star$ - For small t, linear decrease with graph-dependent intercept; for large t, linear increase with intercept independent of the graph topology. Minimum attained is graph-dependent;

- $\rho^\star_{\mathrm{Opt}}$ - For small and large t, respectively, linear decrease and increase with graph-dependent intercept. Minimum attained is independent of graph topology;
- $\rho^\star_{\mathrm{Test}}$ - Linear decrease with graph-dependent intercept up to a threshold independent of the graph topology.

Figure 1 presents $\log - \log$ plots of the Out of Sample Risk against the time horizon t, using the step sizes stated in Corollary 6. All of the behaviours described above, as suggested by our upper bounds, are observed. In particular, recall that our bounds suggest the sub-optimality of the sample rate achieved by the step size aligned with serial SGD ($\rho^\star$), as opposed to the other two choices ($\rho^\star_{\mathrm{Opt}}$ and $\rho^\star_{\mathrm{Test}}$) that depend on the graph topology. Corollary 6 states that the Test Error for $\rho^\star$ yields the rate $\widetilde{O}(n^\alpha/\sqrt{nm})$, as opposed to the rate $\widetilde{O}(1/\sqrt{nm})$ achieved by the other two choices. The former rate is worse (i.e., larger) than the latter for the cycle ($\alpha = 1$) and the grid ($\alpha = 1/2$), while it is of the same order for the complete graph ($\alpha = 0$). Evidence of this behaviour is observed in Figure 1 for $n = 100$, where the Out of Sample Risk related to the cycle and grid is seen to achieve a lower minimum when the step sizes that account for the graph topology are used.

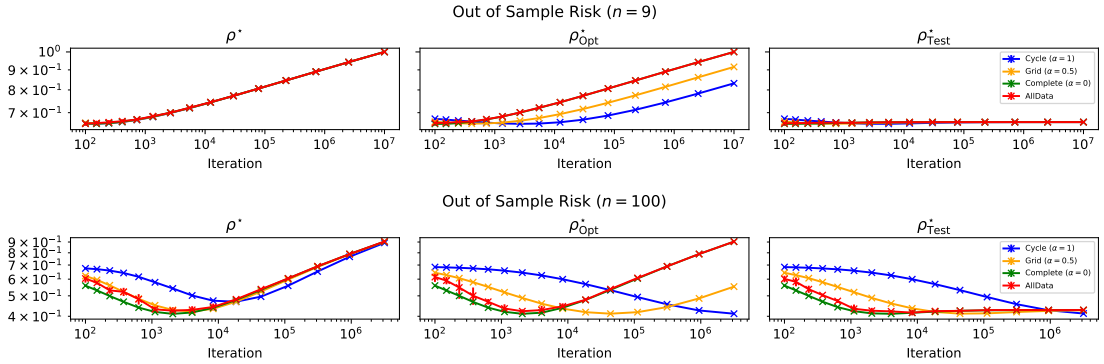


Figure 1: Out of Sample Risk against time horizon for different choices of step size: $\rho^\star$, $\rho^\star_{\mathrm{Opt}}$, and $\rho^\star_{\mathrm{Test}}$. Scales are $\log - \log$. Graph size $n = 9$ (*top*), 100 (*bottom*). Simulations run for 15 values of t from 10^2 to 10^7 (*top*) or $10^{6.5}$ (*bottom*). Each point is an average over 10 (*top*) or 4 (*bottom*) replications with error bars representing 2 standard deviations before taking the log scale (error bars are not visible for large t due to the small variance between repeated runs). *AllData*: serial SGD run on the full data set of 18 (*top*) or 200 (*bottom*) samples with $\rho^\star = \rho^\star_{\mathrm{Opt}} = O(1/\sqrt{t})$ and $\rho^\star_{\mathrm{Test}} = O(1/(\sqrt{t}\sqrt{1+t/m})$. The behaviour of serial SGD is seen to correspond to the behaviour of Distributed SGD on the complete graph, as expected.

6.3. Experimental Results - Large Sample Regime

In this section a larger sample regime ($n = 100$, $m = 25$) is investigated. Due to the number of iterations scaling with the total number of data points i.e. stopping time being of the order $t \sim nm/(1-\sigma_2(P))$, following Remark 8, a fixed step size is used to save running multiple instances of Distributed SGD and save on computational cost. Specifically, the two fixed step size choices considered are: $\rho^\star_{\mathrm{Const}} = O(1/\sqrt{nm})$, to align with serial single-machine SGD; and $\rho^\star_{\mathrm{ConstNet}} = O((1-\sigma_2(P))/\sqrt{nm})$, the step size suggested by Theorem 5 Remark 8 that adjusts for the network

topology. Furthermore, the true underlying optimal parameter $X^{\star\star}$ has its first $\sqrt{d}$ co-ordinates fixed to zero in order to simulate an over parameterised setting. The resulting Out of Sample Risks have been presented within Figure 2.

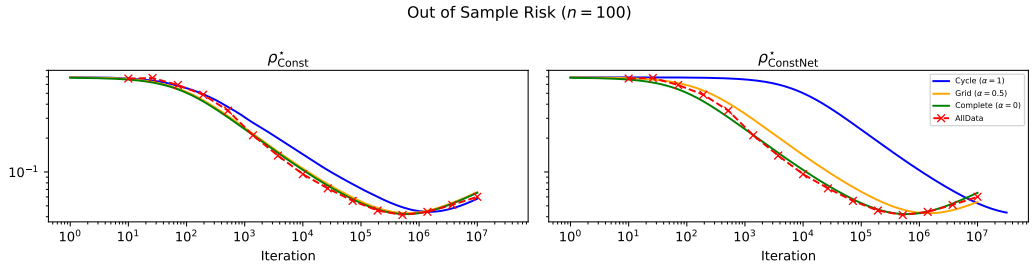


Figure 2: Out of Sample Risk for Distributed SGD with step sizes $\rho^{\star}_{\text{Const}}$ (*Left*) and $\rho^{\star}_{\text{ConstNet}}$ (*Right*) for graph topologies Cycle, Grid and Complete. Each run for 10^7 iterations, while Distributed SGD on cycle topology with $\rho^{\star}_{\text{ConstNet}}$ run for $10^{7.5}$ iterations. Quantity plotted is for a single instance of Distributed SGD. *AllData*: single-machine serial SGD run for 15 different iterations t between 10 and 10^7 with decreasing step size $\rho = O(1/\sqrt{t})$. Both x-axis and y-axis are logarithmic scales.

Firstly, observe that the minimum Out of Sample Risk achieved by Distributed SGD with $\rho^{\star}_{\text{ConstNet}}$ matches the minimum achieved by Serial Single-Machine SGD with decreasing step size (Dashed Red line with markers). Secondly, aligning with the small sample regime in Section 6.2, the minimum out of sample risk (0.0442) for a cycle topology with a constant single-machine serial step size $\rho^{\star}_{\text{Const}}$ is higher than the minimum out of sample risk (0.0436) attained with the constant step size adjusted for the network topology $\rho^{\star}_{\text{ConstNet}}$. We note the simulation for the cycle topology with $\rho^{\star}_{\text{ConstNet}}$ were stopped early at $10^{7.5}$ iterations due to computational cost.

7. Conclusion

We have proposed and investigated graph-dependent implicit regularisation strategies for synchronised Distributed SGD for convex problems in multi-agent learning. Specifically, we have shown how Distributed SGD can retain single-machine serial statistical guarantees by proper tuning of the algorithmic parameters as a function of the graph topology. For convex, Lipschitz, and smooth losses, we showed that Distributed SGD recovers, up to logarithmic terms, the optimal rate of $O(1/\sqrt{nm})$ for single-pass constrained serial SGD (Lan, 2012; Xiao, 2010). For convex and Lipschitz losses, we showed that Distributed SGD recovers, up to logarithmic terms, the best-known rate of $O(1/(nm)^{1/3})$ for single-machine serial SGD with implicit regularisation (Lin et al., 2016b). To obtain these results we: proved Generalisation Error bounds that do not depend on the graph topology and match the bounds in the single-machine serial setting; and derived Optimisation Error bounds that depend on the graph topology. We provided numerical simulations showing that our bounds can be representative of real behaviours.

Our work motivates further investigation of graph-dependent implicit regularisation strategies for decentralised protocols. Since synchronisation and communication are often a dominant bottleneck in distributed computations, further research is needed to investigate the improvement on the communicational and computational complexity that can be obtained by exploiting the interplay between the statistical regularities of the local objective functions and schemes involving mini-

batching, acceleration, and graph sparsification. The latter relates to Gossip protocols where only a random subset of nodes communicate at each iteration (Dimakis et al., 2010). Another direction for future investigation lies in the analysis of adaptive schemes that can contemplate time-dependent step sizes and that can automatically infer the algorithmic parameters of interests, in primis the spectral gap of the communication matrix.

Appendix A. Proofs of Generalisation and Test Error Bounds

This appendix provides the proofs for the Generalisation and Test Error bounds presented within the main body of this paper. First, for completeness, we include the proofs of Proposition 1 and Proposition 12 in Section A.1 and Section A.2, respectively. These results generalise to the distributed setting the Test Error decomposition and the Generalisation Error decomposition via stability used in the single-machine setting, and the proofs follow the exact same arguments as in the single-machine case. Second, we present the proofs of the Generalisation Error bounds for smooth and non-smooth losses in Section A.3 and Section A.4, respectively. For completeness, Section A.3 also includes the proof of stability for the strongly convex case with constraints, which is not covered in the main body but is here presented as it fully generalises the results in Hardt et al. (2016) for Distributed SGD. Third, in Section A.5 we present the proofs of Test Error bounds for smooth and non-smooth losses, referring to Theorem 5 and Theorem 9 within the main body of the work. Finally, in Section A.6 and Section A.7 we give the calculations deriving the rates presented in Corollary 6 and Corollary 10 for smooth and non-smooth losses, respectively. Throughout, we use the notations $\lesssim, \simeq, \gtrsim$, to indicate $\leq, =, \geq$ modulo constants and $\log$ terms.

A.1. Proof of Proposition 1

The proof is analogous to the one given in Hardt et al. (2016) for the single-machine case.

Proof [Proposition 1] We have $r(X_v^t) - r(x^\star) = r(X_v^t) - R(X_v^t) + R(X_v^t) - R(X^\star) + R(X^\star) - r(x^\star)$. Note that $\mathbf{E}R(X^\star) \leq r(x^\star)$, as for any x we have $R(X^\star) \leq R(x)$ so that $\mathbf{E}R(X^\star) \leq \mathbf{E}R(x) = r(x)$, which holds for $x = x^\star$. Thus, $\mathbf{E}\, r(X_v^t) - r(x^\star) \leq \mathbf{E}[r(X_v^t) - R(X_v^t)] + \mathbf{E}[R(X_v^t) - R(X^\star)]$. ■

A.2. Proof of Proposition 12

The proof follows the ideas in Bousquet and Elisseeff (2002) and Hardt et al. (2016) for the single-machine case.

Proof [Proposition 12] As the resampled observation $\widetilde{Z}_{w,k}$ has the same distribution than Z and is independent of both X_v^t and $\mathcal{D}$, we have $\mathbf{E}\, r(X_v^t) = \frac{1}{nm} \sum_{w \in V} \sum_{k=1}^m \mathbf{E}\, \ell(X_v^t, \widetilde{Z}_{w,k})$. As the pair $(X_v^t, Z_{w,k})$ has the same distribution as the pair $(\widetilde{X}(w,k)_v^t, \widetilde{Z}_{w,k})$, the expectation of the empirical risk can be written as $\mathbf{E}R(X_v^t) = \frac{1}{nm} \sum_{w \in V} \sum_{k=1}^m \mathbf{E}\, \ell(\widetilde{X}(w,k)_v^t, \widetilde{Z}_{w,k})$. Thus, $\mathbf{E}[r(X_v^t) - R(X_v^t)] = \frac{1}{nm} \sum_{w \in V} \sum_{k=1}^m \mathbf{E}[\ell(X_v^t, \widetilde{Z}_{w,k}) - \ell(\widetilde{X}(w,k)_v^t, \widetilde{Z}_{w,k})]$. ■

A.3. Proof of Generalisation Error Bounds for Smooth Losses

In this section we prove the Generalisation Error bound presented in Lemma 13 for smooth losses, and we establish a Generalisation Error bound for strongly convex functions. The proof that we present follows the spirit of the proof in Hardt et al. (2016) for the single-machine setting, using algorithmic stability. Specifically, deviations of the algorithm are studied when a single data point in the entire data set is resampled. In the distributed setting that we consider, the training data is spread throughout the communication graph, and we need to consider stability not only with respect to time (i.e., the iteration time of the algorithm), but also with respect to space (i.e., the communication graph). As established in Proposition 12, the Generalisation Error is the average of these deviations. Intermediate steps show that the individual deviations have a dependence on the graph topology, as encoded by the communication matrix P. However, once the average over all deviations is taken, we get results that do not depend on the graph topology.

First, in Section A.3.1 we describe the setup for the stability analysis. Then, in Section A.3.2 we present the proof for the case of convex, Lipschitz, and smooth losses. Finally, in Section A.3.3 we present the case of Lipschitz, smooth, and strongly-convex losses with constraints.

A.3.1. SETUP

For any $w \in V$ and $k \in [m]$, let $\widetilde{\mathcal{D}}(w,k) := \{\mathcal{D} \backslash Z_{w,k}\} \cup \widetilde{Z}_{w,k}$ be the data set in which node w has the k-th observation resampled. Recall that $\widetilde{X}(w,k)_v^t$ denotes the output at node v and time step t of Distributed SGD (1) run with respect to the data set $\widetilde{\mathcal{D}}(w,k)$. From Proposition 12, the link between the Generalisation Error and the ℓ_2 deviation

$$\delta(w,k)_v^t := \|\widetilde{X}(w,k)_v^t - X_v^t\|$$

can be made explicit when the loss function ℓ is L-Lipschitz in the first coordinate (uniformly in the second). Specifically, each term in the double sum $\sum_{k=1}^m \sum_{w \in V}$ in Proposition 12 is bounded by

$$\ell(X_v^t, \widetilde{Z}_{w,k}) - \ell(\widetilde{X}(w,k)_v^t, \widetilde{Z}_{w,k}) \leq L\delta(w,k)_v^t.$$

The results that we derive directly bound the deviation $\delta(w,k)_v^t$. Henceforth, for a given matrix $M \in \mathbb{R}^{n \times n}$ we use the notation M_{vw}^s to represent the quantity $(M^s)_{vw}$, where M^s is the s-th power of M, and the notation M_v to represent the v-th row of M. Hence, for a given vector x, we write $M_v x$ to indicate $\sum_{w \in V} M_{vw} x_w$. For any $x, y \in \mathbb{R}^d$, we let $\langle x, y \rangle = x^\top y = \sum_{i=1}^d x_i y_i$.

Before proceeding to the main proofs we require some standard results relating to the expansive properties of gradient descent updates with smooth and either convex or strongly convex functions. Specifically, for a sufficiently small step size, a result showing that gradient descent updates with smooth and convex function are non-expansive. Meanwhile, for additionally strongly-convex functions, a result showing that gradient descent updates are contractive. The proof can be found in Appendix A of Hardt et al. (2016) and it utilises the co-coercivity of gradients for smooth and convex functions (Nesterov, 2013).

Lemma 17 *Let f be a β-smooth function, convex, and $\eta\beta \leq 2$ with $\eta > 0$. Then, for any $x, y \in \mathbb{R}$,*

$$\|x - y - \eta(\nabla f(x) - \nabla f(y))\| \leq \|x - y\|.$$

Let f be a β-smooth function, γ-strongly convex, and $\eta \leq 2/(\beta + \gamma)$. Then, for any $x, y \in \mathbb{R}$,

$$\|x - y - \eta(\nabla f(x) - \nabla f(y))\| \leq \Big(1 - \frac{\eta\beta\gamma}{\beta + \gamma}\Big)\|x - y\|.$$

A.3.2. CONVEX, LIPSCHITZ, AND SMOOTH LOSSES

We start by stating Proposition 18 that establishes a bound on the deviation $\delta(w,k)_v^t$ that explicitly depends on the graph topology. This is followed by the proof of Lemma 13.

Proposition 18 (Stability for convex, Lipschitz, and smooth losses) *Assume the setting of Lemma 13. Then, for any $v, w \in V, k \in [m]$ and $t \geq 1$,*

$$\mathbf{E}\,\delta(w,k)_v^t = \mathbf{E}\|\widetilde{X}(w,k)_v^t - X_v^t\| \leq \frac{2\eta L}{m}\sum_{s=1}^{t-1} P_{vw}^s.$$

Proof [Proposition 18] Let $\mathcal{F}_1$ be the σ-algebra generated by $\mathcal{D}$ and $\widetilde{\mathcal{D}} := \{\widetilde{\mathcal{D}}(w,k)\}_{w\in V, k\in[m]}$. For any $t \geq 2$, let $\mathcal{F}_t$ be the σ-algebra generated by the data sets $\mathcal{D}$ and $\widetilde{\mathcal{D}}$, and by the collection of uniform random variables $\{K_v^2, \dots, K_v^t\}_{v\in V}$. Plugging the algorithm updates (1) into $\delta(w,k)_v^t$, applying the triangle inequality and using the fact that $\{X_v^{t-1}\}_{v\in V}, \{\widetilde{X}(w,k)_v^{t-1}\}_{v\in V}$, $\mathcal{D}$, and $\widetilde{\mathcal{D}}$ are measurable with respect to $\mathcal{F}_{t-1}$, we get

$$\begin{aligned}
&\mathbf{E}[\delta(w,k)_v^t|\mathcal{F}_{t-1}] \\
&\leq \sum_{l\neq w} P_{vl}\mathbf{E}\Big[\Big\|\widetilde{X}(w,k)_l^{t-1} - X_l^{t-1} - \eta\Big(\nabla\ell(\widetilde{X}(w,k)_l^{t-1}, Z_{l,K_l^t}) - \nabla\ell(X_l^{t-1}, Z_{l,K_l^t})\Big)\Big\|\Big|\mathcal{F}_{t-1}\Big] \quad (4)\\
&\quad + \frac{P_{vw}}{m}\sum_{i\neq k}\Big\|\widetilde{X}(w,k)_w^{t-1} - X_w^{t-1} - \eta\Big(\nabla\ell(\widetilde{X}(w,k)_w^{t-1}, Z_{w,i}) - \nabla\ell(X_w^{t-1}, Z_{w,i})\Big)\Big\| \quad (5)\\
&\quad + \frac{P_{vw}}{m}\Big\|\widetilde{X}(w,k)_w^{t-1} - X_w^{t-1} - \eta\Big(\nabla\ell(\widetilde{X}(w,k)_w^{t-1}, \widetilde{Z}_{w,k}) - \nabla\ell(X_w^{t-1}, Z_{w,k})\Big)\Big\|. \quad (6)
\end{aligned}$$

The above decomposition is in three parts: (4), the terms aligning with agents who do not have a resample datapoint $\forall \ell, \ell \neq w$; (5), the terms at w conditioned on not picking the resample datapoint; and (6), the term at w when picking the resample datapoint. In particular (6) is the only one to involve the difference of two gradients evaluated at different data points ($\widetilde{Z}_{w,k}$ and $Z_{w,k}$). To bound this term, we use the Lipschitz property, $\|\nabla\ell(\,\cdot\,, z)\| \leq L$ for all $z \in \mathcal{Z}$, and get

$$(6) \leq \Big(\delta(w,k)_w^{t-1} + 2\eta L\Big)\frac{P_{vw}}{m}.$$

To bound terms (4) and (5), we use the non-expansive property of the gradient updates arising from the convexity and smoothness of $\ell(\,\cdot\,, z)$, specifically, the inequality $\|x - y - \eta(\nabla\ell(x,z) - \nabla\ell(y,z))\| \leq \|x - y\|$ for $x, y \in \mathbb{R}^d, z \in \mathcal{Z}$ in Lemma 17. In particular we have

$$\begin{aligned}
(4) &\leq \sum_{\ell\neq w} P_{v\ell}\delta(w,k)_\ell^{t-1}\\
(5) &\leq \frac{P_{vw}}{m}\sum_{i\neq k}\delta(w,k)_w^{t-1} = \frac{P_{vw}}{m}(m-1)\delta(w,k)_w^{t-1}.
\end{aligned}$$

This yields

$$\begin{aligned}
\mathbf{E}[\delta(w,k)_v^t|\mathcal{F}_{t-1}] &\leq \sum_{l\neq w} P_{vl}\delta(w,k)_l^{t-1} + \Big(1 - \frac{1}{m}\Big)P_{vw}\delta(w,k)_w^{t-1} + \Big(\delta(w,k)_w^{t-1} + 2\eta L\Big)\frac{P_{vw}}{m}\\
&= \sum_{l\in V} P_{vl}\delta(w,k)_l^{t-1} + \frac{2\eta L}{m}P_{vw}.
\end{aligned}$$

Let $e_v \in \mathbb{R}^n$ be the vector with 1 in the coordinate aligning with node v and 0 everywhere else. Recursively applying the bound above in vector form with $\delta(w,k)^t = \{\delta(w,k)^t_v\}_{v\in V} \in \mathbb{R}^n$ yields (the inequality is meant coordinate-wise)

$$\mathbf{E}\,\delta(w,k)^t = \mathbf{E}[\mathbf{E}[\delta(w,k)^t|\mathcal{F}_{t-1}]] \leq P\,\mathbf{E}\,\delta(w,k)^{t-1} + \frac{2\eta L}{m}Pe_w \leq \frac{2\eta L}{m}\sum_{s=1}^{t-1} P^s e_w,$$

where we used $\delta(w,k)^1_l = \|\widetilde{X}(w,k)^1_l - X^1_l\| = 0$ for all $l \in V$. Recall $(P^s e_w)_v = P^s_{vw}$. ■

The bound in Proposition 18 shows that the expected deviation between the algorithms remains zero until the number of iterations exceeds the natural distance in the graph between node v and node w. This bound naturally reflects the graph topology and captures the propagation of the deviation due to resampling a data point in a specific location of the graph. When combined with the summation over $w \in V$ in Proposition 12, this bound yields a Generalisation Error bound that does not depend on the graph topology: Lemma 13.

Proof [Lemma 13] Plugging the bound from Proposition 18 into the identity from Proposition 12, using that the rows of the matrix P sum to 1, we get

$$\mathbf{E}[r(X^t_v) - R(X^t_v)] \leq \frac{L}{nm}\sum_{w\in V}\sum_{k=1}^{m}\mathbf{E}\,\delta(w,k)^t_v \leq \frac{2\eta L^2}{nm}\sum_{s=1}^{t-1}\sum_{w\in V}P^s_{vw} = \frac{2\eta L^2}{nm}(t-1).$$

■

A.3.3. STRONGLY CONVEX, LIPSCHITZ, AND SMOOTH LOSSES

This section presents a Generalisation Error bound for Distributed SGD when the loss function is strongly convex, smooth, and Lipschitz continuous, generalising the results in Hardt et al. (2016) to the distributed setting. Recall that a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is γ-strongly convex, with $\gamma > 0$, if $f(x) - f(y) \geq \nabla f(y)^\top(x-y) + \gamma\|x-y\|^2/2$ for all $x, y \in \mathbb{R}^d$. As strongly convex functions have unbounded gradients on $\mathbb{R}^d$, we consider the setting where parameters are constrained to be on a compact convex set $\mathcal{X} \subset \mathbb{R}^d$. Let $x \to \Pi(x) = \arg\min_{y\in\mathcal{X}}\|x-y\|$ be the Euclidean projection on $\mathcal{X}$. Then, iteration (1) becomes, for $s \geq 1$,

$$X^{s+1}_v = \Pi\Big(\sum_{w\in V} P_{vw}(X^s_w - \eta\nabla\ell(X^s_w, Z_{w,K^{s+1}_w})\Big). \tag{7}$$

We refer to this variant as Distributed Projected SGD.

To motivate these assumptions, consider the specific case of Tikhonov regularisation, as done in Hardt et al. (2016). If the loss function ℓ is convex, β-smooth, and L-Lipschitz, then the penalised loss function $x \to \ell(x,z) + \frac{\gamma}{2}\|x\|^2$ is γ-strongly convex, $(\beta+\gamma)$-smooth, and $(L+\gamma r)$-Lipschitz when the constraint set is contained in a ball of radius r, i.e., $\mathcal{X} \subseteq \{x \in \mathbb{R}^d : \|x\| \leq r\}$. The next result is the analogue of Lemma 13 with the additional assumption of strong convexity.

Lemma 19 (Generalisation Error bound for strongly-convex, Lipschitz, and smooth losses) *Assume that for any $z \in \mathcal{Z}$ the function $\ell(\,\cdot\,, z)$ is γ-strongly convex, L-Lipschitz, and β-smooth. Let $X_v^1 = 0$ for all $v \in V$. Then, Distributed Projected SGD run on a compact, convex set $\mathcal{X}$ with $\eta \le 2/(\beta+\gamma)$ yields, for any $v \in V$ and $t \ge 1$,*

$$\mathbf{E}[r(X_v^t) - R(X_v^t)] \le \frac{2L^2}{mn}\frac{\beta+\gamma}{\beta\gamma}.$$

Observe that, for a sufficiently small step size $\eta \le 2/(\beta+\gamma)$, the bound obtained is independent of the step size η and number of iterations t. As for the convex and smooth case of Lemma 13, also this bound aligns with the one given in Hardt et al. (2016) for a single agent with nm observations.

The next result is the analogue of Proposition 18.

Proposition 20 (Stability for strongly-convex, Lipschitz, and smooth losses) *Assume the setting of Lemma 19. Then, for any $v, w \in V, k \in [m]$ and $t \ge 1$,*

$$\mathbf{E}\,\delta(w,k)_v^t = \mathbf{E}\,\|\widetilde{X}(w,k)_v^t - X_v^t\| \le \frac{2\eta L}{m}\sum_{s=1}^{t-1}\Big(1 - \frac{\eta\beta\gamma}{\beta+\gamma}\Big)^{s-1} P_{vw}^s.$$

Proof [Proposition 20]

The proof follows the same outline for the proof of Proposition 18. Consider the same setup and notation there defined. Using the non-expansive property of the Euclidean projection, the triangle inequality, and the fact that $\{X_v^{t-1}\}_{v\in V}, \{\widetilde{X}(w,k)_v^{t-1}\}_{v\in V}$, $\mathcal{D}$, and $\widetilde{\mathcal{D}}$ are measurable with respect to $\mathcal{F}_{t-1}$, we get

$$\begin{aligned}
&\mathbf{E}[\delta(w,k)_v^t|\mathcal{F}_{t-1}] \le \mathbf{E}\|\widetilde{X}(w,k)_v^t - X_v^t\| \\
&\le \sum_{l\neq w} P_{vl}\mathbf{E}\Big[\Big\|\widetilde{X}(w,k)_l^{t-1} - X_l^{t-1} - \eta\Big(\nabla\ell(\widetilde{X}(w,k)_l^{t-1}, Z_{l,K_l^t}) - \nabla\ell(X_l^{t-1}, Z_{l,K_l^t})\Big)\Big\|\Big|\mathcal{F}_{t-1}\Big] \quad (8)\\
&\quad + \frac{P_{vw}}{m}\sum_{i\neq k}\Big\|\widetilde{X}(w,k)_w^{t-1} - X_w^{t-1} - \eta\Big(\nabla\ell(\widetilde{X}(w,k)_w^{t-1}, Z_{w,i}) - \nabla\ell(X_w^{t-1}, Z_{w,i})\Big)\Big\| \quad (9)\\
&\quad + \frac{P_{vw}}{m}\Big\|\widetilde{X}(w,k)_w^{t-1} - X_w^{t-1} - \eta\Big(\nabla\ell(\widetilde{X}(w,k)_w^{t-1}, \widetilde{Z}_{w,k}) - \nabla\ell(X_w^{t-1}, Z_{w,k})\Big)\Big\|. \quad (10)
\end{aligned}$$

Term (10) is the only one to involve the difference of two gradients evaluated at different data points ($\widetilde{Z}_{w,k}$ and $Z_{w,k}$). To use the contraction property arising from strong convexity, add and subtract the term $\eta\nabla\ell(\widetilde{X}(w,k)_w^{t-1}, Z_{w,k})$ inside the norm, and use the Lipschitz property to get

$$(10) \le \frac{P_{vw}}{m}\Big\|\widetilde{X}(w,k)_w^{t-1} - X_w^{t-1} - \eta\Big(\nabla\ell(\widetilde{X}(w,k)_w^{t-1}, Z_{w,k}) - \nabla\ell(X_w^{t-1}, Z_{w,k})\Big)\Big\| + \frac{2\eta L}{m}P_{vw}.$$

To bound terms (8) and (9), as well as the bound above for (10), we use the contraction property of the gradient updates from Lemma 17, specifically, the inequality $\|x - y - \eta(\nabla\ell(x,z) - \nabla\ell(y,z))\| \le$

$(1 - \frac{\eta\beta\gamma}{\beta+\gamma})\|x - y\|$ for $x, y \in \mathbb{R}^d$, $z \in \mathcal{Z}$, and $\eta \leq \frac{2}{\beta+\gamma}$. In particular,

$$(8) \leq \Big(1 - \frac{\eta\beta\gamma}{\beta+\gamma}\Big) \sum_{\ell \neq w} P_{v\ell} \delta(w,k)_\ell^{t-1}$$

$$(9) \leq \Big(1 - \frac{\eta\beta\gamma}{\beta+\gamma}\Big) \frac{P_{vw}}{m} \sum_{i \neq k} \delta(w,k)_w^{t-1} = \Big(1 - \frac{\eta\beta\gamma}{\beta+\gamma}\Big) \frac{P_{vw}}{m} (m-1) \delta(w,k)_w^{t-1}$$

$$(10) \leq \Big(1 - \frac{\eta\beta\gamma}{\beta+\gamma}\Big) \frac{P_{vw}}{m} \delta(w,k)_w^{t-1} + \frac{2\eta L}{m} P_{vw}$$

This yields

$$\begin{aligned}
&\mathbf{E}[\delta(w,k)_v^t | \mathcal{F}_{t-1}] \\
&\leq \Big(1 - \frac{\eta\beta\gamma}{\beta+\gamma}\Big) \Big[\sum_{l \neq w} P_{vl} \delta(w,k)_l^{t-1} + \Big(1 - \frac{1}{m}\Big) P_{vw} \delta(w,k)_w^{t-1} + \frac{1}{m} P_{vw} \delta(w,k)_w^{t-1} \Big] + \frac{2\eta L}{m} P_{vw} \\
&= \Big(1 - \frac{\eta\beta\gamma}{\beta+\gamma}\Big) \sum_{l \in V} P_{vl} \delta(w,k)_l^{t-1} + \frac{2\eta L}{m} P_{vw}.
\end{aligned}$$

In vector notation, the above reads

$$\mathbf{E}\, \delta(w,k)^t \leq \Big(1 - \frac{\eta\beta\gamma}{\beta+\gamma}\Big) P\, \mathbf{E}\, \delta(w,k)^{t-1} + \frac{2\eta L}{m} P e_w \leq \frac{2\eta L}{m} \sum_{s=1}^{t-1} \Big(1 - \frac{\eta\beta\gamma}{\beta+\gamma}\Big)^{s-1} P^s e_w$$

where we used $\delta(w,k)_l^1 = \|\widetilde{X}(w,k)_l^1 - X_l^1\| = 0$ for all $l \in V$ and recursively applied the above bound to $\mathbf{E}[\delta(w,k)^t]$. ■

With Proposition 20 in hand, we prove Lemma 19.

Proof [Lemma 19] Plugging the bound from Proposition 20 into the identity from Proposition 12, using that the rows of the matrix P sum to 1, we get

$$\mathbf{E}[r(X_v^t) - R(X_v^t)] \leq \frac{L}{nm} \sum_{w \in V} \sum_{k=1}^{m} \mathbf{E}\, \delta(w,k)_v^t \leq \frac{2\eta L^2}{mn} \sum_{s=1}^{t-1} \Big(1 - \frac{\eta\beta\gamma}{\beta+\gamma}\Big)^{s-1},$$

and the proof is concluded by summing the geometric projection for t going to infinity, using that the assumption $\eta \leq \frac{2}{\beta+\gamma}$ implies that $\frac{\eta\beta\gamma}{\beta+\gamma} < 1$. ■

A.4. Proof of Generalisation Error Bound for Non-Smooth Losses

This section presents Generalisation Error bounds for Distributed SGD when losses are assumed to be non-smooth, aligning with Lemma 14 within the main body of the text. In this case we follow the approach in (Lin et al., 2016a, Appendix B) that involves controlling the Generalisation Error by using standard Rademacher complexity's arguments through Assumption 1 *(b)* and bounding the norm of the iterates through Assumption 1 *(a)*. We start by presenting Lemma 21 which bounds the iterates produced by the Distributed SGD. This is followed by the proof for the Generalisation Error bound for non-smooth losses Lemma 14.

Lemma 21 *Assume there exist $C \leq B$ such that for each $z \in \mathcal{Z}$ the function $\ell(\,\cdot\,, z)$ is convex, L-Lipschitz, bounded above at zero $\ell(0, z) \leq B$, and bound uniformly from below $\ell(x, z) \geq C$ for $x \in \mathbb{R}^d$. Let $X_v^1 = 0$ for all $v \in V$. Then, Distributed SGD yields, for any $v \in V$ and $t \geq 1$,*

$$\|X_v^t\| \leq \sqrt{(t-1)(\eta^2 L^2 + 2\eta(B-C))}.$$

Proof Let $x \in \mathbb{R}^d$. By the Distributed SGD update (1) we get

$$\|X_v^t - x\| \leq \sum_{w \in V} P_{vw} \|X_w^{t-1} - \eta \partial \ell(X_w^{t-1}, Z_{w,K_w^t}) - x\|. \tag{11}$$

The convexity of $\ell(\,\cdot\,, z)$ yields

$$\langle \partial \ell(X_w^{t-1}, Z_{w,K_w^t}), x - X_w^{t-1} \rangle \leq \ell(x, Z_{w,K_w^t}) - \ell(X_w^{t-1}, Z_{w,K_w^t}),$$

and the Lipschitz continuity of $\ell(\,\cdot\,, z)$ yields $\|\partial \ell(X_w^{t-1}, Z_{w,K_w^t})\| \leq L$. Thus,

$$\begin{aligned}
&\|X_w^{t-1} - \eta \partial \ell(X_w^{t-1}, Z_{w,K_w^t}) - x\|^2 \\
&= \|X_w^{t-1} - x\|^2 + \eta^2 \|\partial \ell(X_w^{t-1}, Z_{w,K_w^t})\|^2 + 2\eta \langle \partial \ell(X_w^{t-1}, Z_{w,K_w^t}), x - X_w^{t-1} \rangle \\
&\leq \|X_w^{t-1} - x\|^2 + \eta^2 L^2 + 2\eta(\ell(x, Z_{w,K_w^t}) - \ell(X_w^{t-1}, Z_{w,K_w^t})).
\end{aligned}$$

Setting $x = 0$, using that $\ell(X_w^{t-1}, Z_{w,K_w^t}) \geq C$ as well as the assumption $\ell(0, Z_{w,K_w^t}) \leq B$, we get

$$\|X_w^{t-1} - \eta \partial \ell(X_w^{t-1}, Z_{w,K_w^t})\|^2 \leq \|X_w^{t-1}\|^2 + \eta^2 L^2 + 2\eta(B - C).$$

Using that the matrix P is doubly stochastic, the bound (11) yields the recursion

$$\max_{v \in V} \|X_v^t\|^2 \leq \max_{w \in V} \|X_w^{t-1} - \eta \partial \ell(X_w^{t-1}, Z_{w,K_w^t})\|^2 \leq \max_{v \in V} \|X_v^{t-1}\|^2 + \eta^2 L^2 + 2\eta(B - C),$$

so recursively applying the above bound and taking square root gives

$$\|X_v^t\| \leq \max_{v \in V} \|X_v^t\| \leq \sqrt{(t-1)(\eta^2 L^2 + 2\eta(B-C))}.$$

■

Proof [Lemma 14] Standard Rademacher complexity's arguments utilising the symmetrisation technique and Assumption 1 *(b)* yield, for any $\widetilde{\mathcal{X}} \subseteq \mathcal{X}$,

$$\mathbf{E} \sup_{x \in \widetilde{\mathcal{X}}} (r(x) - R(x)) \leq 2\mathbf{E} \sup_{x \in \widetilde{\mathcal{X}}} \frac{1}{nm} \sum_{i=1}^{nm} \sigma_i \ell(x, z_i) \leq 2D \frac{\sup_{x \in \widetilde{\mathcal{X}}} \|x\|}{\sqrt{nm}}.$$

By Lemma 21 we know that the iterates are contained in the ball $\widetilde{\mathcal{X}} = \{x \in \mathbb{R}^d : \|x\| \leq \sqrt{(t-1)(\eta^2 L^2 + 2\eta(B-C))}\}$, so that

$$\mathbf{E}[r(X_v^t) - R(X_v^t)] \leq \mathbf{E} \sup_{x \in \widetilde{\mathcal{X}}} (r(x) - R(x)) \leq 2D \sqrt{\frac{(t-1)(\eta^2 L^2 + 2\eta(B-C))}{nm}}.$$

■

A.5. Proof of Test Error Bounds for Smooth and Non-Smooth Losses

This section gives the proofs of the Test Error bounds presented within the main body of the work, namely Theorem 5 for convex, Lipschitz, and smooth losses, and Theorem 9 for convex and Lipschitz losses. This is achieved by using the error decomposition given in Proposition 1, and by bringing together the Generalisation Error bounds and the Optimisation Error bounds in Section 5.

Proof [Theorem 5] By the convexity of the Test Risk r, using Proposition 1, we get

$$\mathbf{E}\, r\Big(\frac{1}{t}\sum_{s=1}^{t} X_v^{s+1}\Big) - r(x^\star) \leq \frac{1}{t}\sum_{s=1}^{t}\Big(\underbrace{\mathbf{E}[r(X_v^{s+1}) - R(X_v^{s+1})]}_{\text{Generalisation Error}} + \underbrace{\mathbf{E}[R(X_v^{s+1}) - R(X^\star)]}_{\text{Optimisation Error}}\Big).$$

The proof follows by applying Lemma 13 for the Generalisation Error, which yields

$$\frac{1}{t}\sum_{s=1}^{t}\mathbf{E}[r(X_v^{s+1}) - R(X_v^{s+1})] \leq \frac{2\eta L^2}{nm}\frac{1}{t}\sum_{s=1}^{t} s = \frac{\eta L^2}{nm}(t+1),$$

and by the Optimisation Error bound from Lemma 15. ■

Proof [Theorem 9] By the convexity of the Test Risk r, using Proposition 1, we get

$$\mathbf{E}\, r\Big(\frac{1}{t}\sum_{s=1}^{t} X_v^{s}\Big) - r(x^\star) \leq \frac{1}{t}\sum_{s=1}^{t}\Big(\underbrace{\mathbf{E}[r(X_v^{s}) - R(X_v^{s})]}_{\text{Generalisation Error}} + \underbrace{\mathbf{E}[R(X_v^{s}) - R(X^\star)]}_{\text{Optimisation Error}}\Big).$$

The proof follows by applying Lemma 14 for the Generalisation Error, which yields

$$\frac{1}{t}\sum_{s=1}^{t}\mathbf{E}[r(X_v^{s}) - R(X_v^{s})] \leq 2D\sqrt{\frac{(t-1)(\eta^2L^2 + 2\eta(B-C))}{nm}},$$

and by the Optimisation Error bound from Lemma 16. ■

A.6. Calculations for Corollary 6 (Convex, Lipschitz, and Smooth)

This section presents the calculations needed to populate the table of rates in Corollary 6 in the case of convex, Lipschitz, and smooth losses. The simplification $1/(\beta + 1/\rho) \leq \rho$ is used. Additionally, minimisations are performed up to first-order terms in ρ, possibly neglecting logarithmic terms. This section is divided into four parts:

- **Optimisation Error** calculates the step size $\rho^\star_{\text{Opt}}$ minimising the Optimisation Error bound;
- **Test Error** calculates the step size $\rho^\star_{\text{Test}}$ that minimises the Test Error bound;
- **Early Stopping Optimisation** calculates the number of iterations that minimises the Test Error bound when the step size $\rho^\star_{\text{Opt}}$ is used;
- **Early Stopping Single-Machine Serial** calculates the number of iterations that minimises the Test Error bound when the step size $\rho^\star = O(1/\sqrt{t})$ is used.

Optimisation Error. Optimising over first-order terms in ρ in the Optimisation Error bound of Lemma 15 with $1/(\beta+1/\rho) \leq \rho$ we get

$$\rho^\star_{\mathrm{Opt}} = \mathrm{argmin}_\rho \Big\{ \frac{\rho}{2}\sigma^2 + \frac{G^2}{2t\rho} + 3L\kappa\rho \frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)} \Big\} = \frac{G}{\sqrt{t}} \frac{1}{\sqrt{6L\kappa \frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)} + \sigma^2}},$$

which yields with $\frac{3+\beta\rho}{\beta+1/\rho} \leq 4\rho$ from $3/(\beta+1/\rho) \leq 3\rho$ and $\beta/(\beta+1/\rho) \leq \rho$ the Optimisation Error bound

$$\begin{aligned}
&\mathbf{E}\Big[R\Big(\frac{1}{t}\sum_{s=1}^{t} X_v^{s+1}\Big) - R(X^\star)\Big] \\
&\leq \frac{G}{\sqrt{t}}\sqrt{6L\kappa\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)} + \sigma^2} + \frac{\beta G^2}{2t} + 18\kappa^2\beta\rho^2_{\mathrm{Opt}}\Big(\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)}\Big)^2 \\
&\leq \frac{G}{\sqrt{t}}\sqrt{6L\kappa\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)} + \sigma^2} + \frac{\beta G^2}{2t}\Big[1 + \frac{6\kappa}{L}\frac{\Big(\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)}\Big)^2}{\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)} + \frac{\sigma^2}{6L\kappa}}\Big].
\end{aligned}$$

This bound is $\widetilde{O}(1/\sqrt{(1-\sigma_2(P))t})$ as the second term is $\widetilde{O}(1/((1-\sigma_2(P))t))$.

Test Error. Consider the Test Error bound in Theorem 5 with $1/(\beta+1/\rho) \leq \rho$. Optimising over first-order terms in ρ we get

$$\begin{aligned}
\rho^\star_{\mathrm{Test}} &= \mathrm{argmin}_\rho \Big\{ \frac{\rho}{2}\sigma^2 + \frac{G^2}{2t\rho} + 3L\kappa\rho\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)} + \frac{\rho L^2}{nm}(t+1)\Big\} \\
&= \frac{G}{\sqrt{t}}\frac{1}{\sqrt{6L\kappa\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)} + \sigma^2 + \frac{2L^2(t+1)}{nm}}},
\end{aligned}$$

which yields with with $\frac{3+\beta\rho}{\beta+1/\rho} \leq 4\rho$ from $3/(\beta+1/\rho) \leq 3\rho$ and $\beta/(\beta+1/\rho) \leq \rho$ the Test Error bound

$$\begin{aligned}
&\mathbf{E}\, r\Big(\frac{1}{t}\sum_{s=1}^{t} X_v^{s+1}\Big) - r(x^\star) \\
&\leq \frac{G}{\sqrt{t}}\sqrt{6L\kappa\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)} + \sigma^2 + \frac{2L^2}{nm}(t+1)} + \frac{\beta G^2}{2t} + 18\kappa^2\beta\rho^2_{\mathrm{Test}}\Big(\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)}\Big)^2 \\
&\leq \frac{G}{\sqrt{t}}\sqrt{6L\kappa\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)} + \sigma^2 + \frac{2L^2}{nm}(t+1)} \\
&\qquad + \frac{\beta G^2}{2t}\Big[1 + \frac{6\kappa}{L}\frac{\Big(\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)}\Big)^2}{\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)} + \frac{1}{6L\kappa}(\sigma^2 + \frac{2L^2}{nm}(t+1))}\Big].
\end{aligned}$$

This bound is $\widetilde{O}\Big(\sqrt{\frac{1}{t(1-\sigma_2(P))} + \frac{1}{nm}}\Big)$ as the second term is $\widetilde{O}(1/((1-\sigma_2(P))t))$. This is $\widetilde{O}(\frac{1}{\sqrt{nm}})$ when $t \gtrsim nm/(1-\sigma_2(P))$.

Early Stopping Optimisation. Considering the Test Error bound from Theorem 5 with step size $\rho = \rho^{\star}_{\mathrm{Opt}}$ and $1/(\beta + 1/\rho) \leq \rho$ we get

$$\mathbf{E}\, r\Big(\frac{1}{t}\sum_{s=1}^{t} X_v^{s+1}\Big) - r(x^\star) \leq G\Bigg[\frac{1}{\sqrt{t}}\sqrt{6L\kappa\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)} + \sigma^2} + \frac{2L^2\sqrt{t}}{nm}\sqrt{\frac{1}{6L\kappa\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)} + \sigma^2}}\Bigg]$$
$$+ \frac{\beta G^2}{2t}\Big[1 + \frac{6\kappa}{L}\frac{\Big(\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)}\Big)^2}{\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)} + \frac{\sigma^2}{6L\kappa}}\Big],$$

where $(t+1)/\sqrt{t} \leq 2\sqrt{t}$ was used. The first term is dominant and $O\Big(\sqrt{\frac{\log(t\sqrt{n})}{t(1-\sigma_2(P))}} + \frac{1}{nm}\sqrt{\frac{t(1-\sigma_2(P))}{\log(t\sqrt{n})}}\Big)$ while the second term is $\widetilde{O}(1/(1-\sigma_2(P)t))$. To minimise the first term with respect to t, consider the more tractable form

$$\frac{1}{\sqrt{t}}\sqrt{6L\kappa\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)} + \sigma^2} + \frac{2L^2\sqrt{t}}{nm}\sqrt{\frac{1}{6L\kappa\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)} + \sigma^2}}$$
$$\leq \frac{\sigma}{\sqrt{t}} + \sqrt{6L\kappa\frac{\log((t+1)\sqrt{n})}{t(1-\sigma_2(P))}} + \frac{2L^2}{nm}\sqrt{\frac{t(1-\sigma_2(P))}{6L\kappa\log((t+1)\sqrt{n})}}.$$

An approximate minimiser in t neglecting the $\log((t+1)\sqrt{n})$ in the denominator is given by

$$\frac{t}{\log((t+1)\sqrt{n})} = \mathrm{argmin}_{c\geq 0}\left\{\sqrt{\frac{6L\kappa}{c(1-\sigma_2(P))}} + \frac{2L^2}{nm}\sqrt{\frac{c(1-\sigma_2(P))}{6L\kappa}}\right\} = 3\frac{\kappa}{L}\frac{nm}{1-\sigma_2(P)}.$$

This choice yields the Test Error bound

$$\mathbf{E}\, r\Big(\frac{1}{t}\sum_{s=1}^{t} X_v^{s+1}\Big) - r(x^\star) \leq \frac{G}{\sqrt{nm}}\Big[\sigma\sqrt{\frac{L(1-\sigma_2(P))}{3\kappa}} + 2\sqrt{2}L\Big]$$
$$+ \frac{\beta G^2 L(1-\sigma_2(P))}{6\kappa nm}\Big[1 + \frac{6\kappa}{L}\frac{\Big(\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)}\Big)^2}{\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)} + \frac{\sigma^2}{6L\kappa}}\Big]$$

which is a $\widetilde{O}(\frac{1}{\sqrt{nm}})$ Test Error bound obtained with $t \simeq nm/(1-\sigma_2(P))$ iterations.

Early Stopping Single-Machine Serial. Considering the Test Error bound of Theorem 5 with $1/\beta + 1/\rho \leq \rho$ and $\rho = \rho^\star = \frac{G}{Lc\sqrt{t}}$ for some constant $c > 0$. Plugging in we get

$$\mathbf{E}\, r\Big(\frac{1}{t}\sum_{s=1}^{t} X_v^{s+1}\Big) - r(x^\star) \leq \frac{G}{c}\Big[3\kappa\frac{\log((t+1)\sqrt{n})}{(1-\sigma_2(P))\sqrt{t}} + \frac{2L}{nm}\sqrt{t}\Big]$$
$$+ \frac{G}{2\sqrt{t}}\Big(\frac{\sigma^2}{cL} + cL\Big) + \frac{\beta G^2}{2t}\Big[1 + \frac{9(3+\beta\rho)\kappa^2}{c^2L^2}\frac{\log^2((t+1)\sqrt{n})}{(1-\sigma_2(P))^2}\Big],$$

where $(t+1)/\sqrt{t} \leq 2\sqrt{t}$ for $t \geq 1$ was used on the Generalisation Error bound. The above bound is dominated by the first term which is $\widetilde{O}\Big(\frac{1}{(1-\sigma_2(P))\sqrt{t}} + \frac{\sqrt{t}}{nm}\Big)$. Minimising up to log terms yields

$$t = \frac{3\kappa}{2L}\frac{nm}{1-\sigma_2(P)}.$$

This choice yields the Test Error bound

$$\mathbf{E}\, r\Big(\frac{1}{t}\sum_{s=1}^{t} X_v^{s+1}\Big) - r(x^\star) \leq \frac{G}{c}\sqrt{\frac{6\kappa L}{nm(1-\sigma_2(P))}}\Big[\log((t+1)\sqrt{n}) + 1\Big]$$
$$+ \sqrt{\frac{L(1-\sigma_2(P))}{6\kappa nm}}\Big(\frac{\sigma^2}{cL} + cL\Big) + \frac{L\beta G^2(1-\sigma_2(P))}{3\kappa nm}\Big[1 + \frac{9(3+\beta\rho)\kappa^2}{c^2L^2}\frac{\log^2((t+1)\sqrt{n})}{(1-\sigma_2(P))^2}\Big],$$

which is dominated by the first term that is $\widetilde{O}\Big(\frac{1}{\sqrt{nm(1-\sigma_2(P))}}\Big)$, as the third term is $\widetilde{O}\Big(\frac{1}{nm(1-\sigma_2(P))}\Big)$. Regarding the choice of constant c, note the above is decreasing up to $c = (1-\sigma_2(P))^{-1/2}$, in which case the $O(1/\sqrt{nm})$ rate for $\rho^\star_{\mathrm{Opt}}$ is recovered.

A.7. Calculations for Corollary 10 (Convex and Lipschitz)

This section presents the calculations needed to populate the table of rates in Corollary 10 in the case of convex and Lipschitz losses. This section is divided into four parts:

- **Optimisation Error** calculates the step size $\eta^\star_{\mathrm{Opt}}$ minimising the Optimisation Error bound;
- **Test Error** calculates the step size $\eta^\star_{\mathrm{Test}}$ that minimises the Test Error bound;
- **Early Stopping Optimisation** calculates the number of iterations that minimises the Test Error bound when the step size $\eta^\star_{\mathrm{Opt}}$ is used;
- **Early Stopping Single-Machine Serial** calculates the number of iterations that minimises the Test Error when the step size $\eta^\star = O(1/\sqrt{t})$ is used.

Optimisation Error. Minimising the Optimisation Error bound in Lemma 16 with respect to the step size yields $\eta = \eta^\star_{\mathrm{Opt}} = \frac{G}{L\sqrt{19t}}\sqrt{\frac{1-\sigma_2(P)}{\log(t\sqrt{n})}}$ and

$$\mathbf{E}\Big[R\Big(\frac{1}{t}\sum_{s=1}^{t} X_v^s\Big) - R(X^\star)\Big] \leq \sqrt{19}\frac{GL}{\sqrt{t}}\sqrt{\frac{\log(t\sqrt{n})}{1-\sigma_2(P)}}.$$

Test Error. In this section the step size

$$\eta = \eta^\star_{\mathrm{Test}} = \frac{G}{L\sqrt{t}}\frac{1}{\sqrt{\frac{19}{2}\frac{\log(t\sqrt{n})}{1-\sigma_2(P)} + \frac{t}{(nm)^{2/3}}}}$$

is shown to ensure that the Test Error bound in Theorem 9 converges in a time uniform manner to a quantity of order $\widetilde{O}(1/(nm)^{1/3})$. We consider the Optimisation and Generalisation Error separately.

The Optimisation Error bound with this step size yields

$$\frac{19}{2}\frac{\eta^\star_{\text{Test}}L^2\log(t\sqrt{n})}{1-\sigma_2(P)}+\frac{G^2}{2\eta^\star_{\text{Test}}t}=\frac{GL}{\sqrt{t}}\sqrt{\frac{19}{2}\frac{\log(t\sqrt{n})}{1-\sigma_2(P)}+\frac{t}{(nm)^{2/3}}}\Big[\frac{\frac{19}{2}\frac{\log(t\sqrt{n})}{1-\sigma_2(P)}}{\frac{19}{2}\frac{\log(t\sqrt{n})}{1-\sigma_2(P)}+\frac{t}{(nm)^{2/3}}}+\frac{1}{2}\Big]$$
$$\leq\frac{3}{2}\frac{GL}{\sqrt{t}}\sqrt{\frac{19}{2}\frac{\log(t\sqrt{n})}{1-\sigma_2(P)}+\frac{t}{(nm)^{2/3}}},$$

which is $\widetilde{O}(\frac{1}{(nm)^{1/3}})$ when the number of iterations satisfies $t\geq\frac{19}{2}\log(t\sqrt{n})(nm)^{2/3}/(1-\sigma_2(P))$. We split the Generalisation Error bound term into two parts

$$2D\sqrt{\frac{(t-1)(\eta^2L^2+2\eta(B-C))}{nm}}\leq 2\eta DL\sqrt{\frac{t}{nm}}+2\sqrt{2}D\sqrt{\frac{\eta t(B-C)}{nm}}\tag{12}$$

and bounded each part separately. The first quantity in (12) with the step size $\eta=\eta^\star_{\text{Test}}$ becomes

$$2\eta^\star_{\text{Test}}DL\sqrt{\frac{t}{nm}}=\frac{GD}{\sqrt{nm}}\frac{1}{\sqrt{\frac{19}{2}\frac{\log(t\sqrt{n})}{1-\sigma_2(P)}+\frac{t}{(nm)^{2/3}}}}\leq\frac{GD}{\sqrt{nm}}\sqrt{\frac{2}{19}\frac{1-\sigma_2(P)}{\log(t\sqrt{n})}},$$

which is $O(1/\sqrt{nm})$, and thus $O(1/(nm)^{1/3})$. For the second quantity in (12), its square yields

$$\begin{aligned}8D^2\frac{\eta^\star_{\text{Test}}t(B-C)}{nm}&=8D^2\frac{(B-C)G}{Lnm}\sqrt{\frac{t}{\frac{19}{2}\frac{\log(t\sqrt{n})}{1-\sigma_2(P)}+\frac{t}{(nm)^{2/3}}}}\\&=8D^2\frac{(B-C)GL}{L(nm)^{2/3}}\sqrt{\frac{t}{\frac{19}{2}\frac{\log(t\sqrt{n})(nm)^{2/3}}{1-\sigma_2(P)}+t}}\\&\leq 8D^2\frac{(B-C)G}{L(nm)^{2/3}}.\end{aligned}$$

Therefore, when using step size $\eta^\star_{\text{Test}}$ with $t\gtrsim(nm)^{2/3}/(1-\sigma_2(P))$ the Test Error is bounded by the sum of three quantities each of which are $\widetilde{O}(1/(nm)^{1/3})$.

Early Stopping Optimisation. Setting $\eta=\eta^\star_{\text{Opt}}$ in the Test Error bound in Theorem 9 and using (12) to split the Generalisation Error we get

$$\begin{aligned}\mathbf{E}\,r\Big(\frac{1}{t}\sum_{s=1}^{t}X_v^s\Big)-r(x^\star)\leq\sqrt{19}\frac{GL}{\sqrt{t}}\sqrt{\frac{\log(t\sqrt{n})}{1-\sigma_2(P)}}&\\+\frac{2GD}{\sqrt{19nm}}\sqrt{\frac{1-\sigma_2(P)}{\log(t\sqrt{n})}}+2\sqrt{2}D\sqrt{\frac{G(B-C)}{Lnm}}\sqrt{\frac{t(1-\sigma_2(P))}{19\log(t\sqrt{n})}}.&\end{aligned}$$

This is $O\Big(\sqrt{\frac{\log(t\sqrt{n})}{t(1-\sigma_2(P))}}+\sqrt{\frac{1}{nm}\sqrt{\frac{t(1-\sigma_2(P))}{\log(t\sqrt{n})}}}\Big)$ as the second term is dominated by the first and third. Neglecting the $\log(t\sqrt{n})$ term and approximately minimising in t yields

$$\begin{aligned}\frac{t}{\log(t\sqrt{n})} &= \operatorname{argmin}_{c>0}\Big\{GL\sqrt{\frac{1}{c}\frac{19}{1-\sigma_2(P)}}+2\sqrt{2}D\sqrt{\frac{G(B-C)}{Lnm}\sqrt{c\frac{1-\sigma_2(P)}{19}}}\Big\}\\ &= \frac{19L^2(Gnm)^{2/3}}{(1-\sigma_2(P))(2(B-C))^{2/3}D^{4/3}}\end{aligned}$$

with the final bound

$$\mathbf{E}\, r\Big(\frac{1}{t}\sum_{s=1}^{t}X_v^s\Big)-r(x^\star)\le 2^{1/3}3\frac{G^{2/3}(B-C)^{1/3}D^{2/3}}{(nm)^{1/3}}+\frac{2GD}{\sqrt{19nm}}\sqrt{\frac{1-\sigma_2(P)}{\log(t\sqrt{n})}}.$$

This is a $O(1/(nm)^{1/3})$ Test Error bound obtained with $t\simeq(nm)^{2/3}/(1-\sigma_2(P))$ iterations.

Early Stopping Single-Machine Serial. Setting $\eta=\eta^\star=\frac{G}{L\sqrt{19t}}$ in the Test Error bound in Theorem 9 and using (12) to split the Generalisation Error gives

$$\mathbf{E}\, r\Big(\frac{1}{t}\sum_{s=1}^{t}X_v^s\Big)-r(x^\star)\le GL\sqrt{\frac{19}{t}}\frac{\log(t\sqrt{n})}{1-\sigma_2(P)}+\frac{2GD}{\sqrt{19nm}}+2\sqrt{2}D\sqrt{\frac{G(B-C)}{Lnm}\sqrt{\frac{t}{19}}},$$

which is $\widetilde{O}\Big(\frac{1}{(1-\sigma_2(P))\sqrt{t}}+\sqrt{\frac{1}{nm}\sqrt{t}}\Big)$ as the second term is dominated by the first and third. Neglecting the $\log(t\sqrt{n})$ term and approximately minimising the above with respect to the number of iterations t yields

$$\begin{aligned}t &= \operatorname{argmin}_{c>0}\Big\{GL\sqrt{\frac{19}{c}}\frac{\log(t\sqrt{n})}{1-\sigma_2(P)}+2\sqrt{2}D\sqrt{\frac{G(B-C)}{Lnm}\sqrt{\frac{c}{19}}}\Big\}\\ &= \frac{19\log^{4/3}(t\sqrt{n})(Gnm)^{2/3}L^2}{(1-\sigma_2(P)^{4/3}(2(B-C))^{2/3}D^{4/3}}\end{aligned}$$

with the resulting bound

$$\mathbf{E}\, r\Big(\frac{1}{t}\sum_{s=1}^{t}X_v^s\Big)-r(x^\star)\le 2^{1/3}3\Big(\frac{\log(t\sqrt{n})}{1-\sigma_2(P)}\Big)^{1/3}\frac{G^{2/3}(B-C)^{1/3}D^{2/3}}{(nm)^{1/3}}+\frac{2GD}{\sqrt{19nm}}.$$

This is $\widetilde{O}(1/(nm(1-\sigma_2(P))^{1/3})$ and is obtained with $t\simeq(nm)^{2/3}/(1-\sigma_2(P))^{4/3}$ iterations.

A.8. Calculation for Remark 8

In this section it is shown that Distributed SGD with step size choice $\rho=O((1-\sigma_2(P))/\sqrt{nm})$ and iterations $t=O(nm/(1-\sigma_2(P)))$ achieves optimal statistical rates up to logarithmic factors for convex, smooth and Lipschitz losses.

Begin by plugging $\rho = G(1-\sigma_2(P))/(L\sqrt{nm})$ into the Test Error bound of Theorem 5 with $1/(\beta+1/\rho) \leq \rho$ and $(3+\beta\rho)/(\beta+1/\rho) \leq 4\rho$, the latter arising from $3/(\beta+1/\rho) \leq 3\rho$ and $\beta\rho/(\beta+1/\rho) \leq \rho$. This then yields the Test Error bound

$$\mathbf{E}\, r\Big(\frac{1}{t}\sum_{s=1}^{t} X_v^{s+1}\Big) - r(x^\star) \leq \frac{2(1-\sigma_2(P))GLt}{(nm)^{3/2}} + \frac{\sigma^2(1-\sigma_2(P))G}{2L\sqrt{nm}} + \frac{\beta G^2}{2t} + \frac{GL\sqrt{nm}}{2t(1-\sigma_2(P))} + \frac{3G\kappa\log((t+1)\sqrt{n})}{\sqrt{nm}} + 18\frac{\beta G^2\log^2((t+1)\sqrt{n})}{L^2 nm}.$$

Choosing $t = (nm)/(1-\sigma_2(P))$ we see that the first and fourth terms become $O(1/\sqrt{nm})$ while the remaining terms are in this case $\widetilde{O}(1/\sqrt{nm})$.

Appendix B. Proofs of Optimisation Error bounds

This appendix presents Optimisation Error bounds for the Distributed Stochastic Subgradient Descent algorithm. Here we consider the general setting of stochastic first-order oracles. The Optimisation Error bounds presented within the main body of this work, specifically Lemma 15 and Lemma 16 for smooth and non-smooth losses, follow from Corollary 27 and Corollary 25 within this appendix.

B.1. Setup

Let (V, E) be a simple undirected graph with n nodes, and let $P \in \mathbb{R}^{n\times n}$ be a doubly stochastic matrix supported on the graph, i.e., $P_{ij} \neq 0$ only if $\{i,j\} \in E$. For each $v \in V$, let $F_v : \mathbb{R}^d \to \mathbb{R}$ be a random convex function. We consider the problem of minimizing the function $x \to \overline{F}(x) := \frac{1}{n}\sum_{v\in V} F_v(x)$ over $x \in \mathbb{R}^d$. Let $X^\star$ be a minimum of $\overline{F}$. Assume that $\mathbf{E}[\|X^\star\|^2] \leq G^2$ for a positive constant G. Given the initial vectors $\{X_v^1 = 0\}_{v\in V}$, throughout this appendix, we consider the following update for $s \geq 1$:

$$X_v^{s+1} = \sum_{w\in V} P_{vw}(X_w^s - \eta G_w^{s+1}), \tag{13}$$

where $\eta > 0$ is the step size, and each $G_v^{s+1} \in \mathbb{R}^d$ is an estimator of the subgradient of F_v evaluated at X_v^s. Specifically, for each $s \geq 1$ let $\mathcal{F}_s$ be the σ-algebra generated by the random functions $\{F_v\}_{v\in V}$ and by the estimators $\{G_v^k\}_{k\in\{2,\dots,s\}}$. We have, for any $s \geq 1$, $v \in V$,

$$\mathbf{E}[G_v^{s+1}|\mathcal{F}_s] \in \partial F_v(X_v^s). \tag{14}$$

Note that both $\{X_v^s\}_{v\in V}$ and $X^\star$ are measurable with respect to $\mathcal{F}_s$. Assume, for any $s \geq 1$, $v \in V$,

$$\mathbf{E}[\|G_v^{s+1}\|^2|\mathcal{F}_s] \leq L^2. \tag{15}$$

Section B.2 presents results for the setting just introduced under the additional assumption that the functions $\{F_v\}_{v\in V}$ are L-Lipschitz. Section B.3 presents results for the case where the functions $\{F_v\}_{v\in V}$ are smooth (Lipschitz continuity is not assumed in this case). Finally, Section B.4 checks that the assumptions of this general setting are satisfied for the specific case of algorithm (1).

The bounds that we derive are proved controlling the deviation of the output of the algorithm X_v^s from its network average $\overline{X}^s := \frac{1}{n}\sum_{v\in V} X_v^s$ on the one hand (*network term*), and bounding the deviation of $\overline{X}^s$ from the solution $X^\star$ on the other end (*optimisation term*). This strategy was originally proposed in Nedić et al. (2009) and used in Duchi et al. (2012) to get bounds that depend on the graph topology for a dual method in constrained optimisation. In the smooth case, we present a bound that also depends on the noise of the gradient (*gradient noise term*).

Remark 22 *The bounds that we derive naturally generalise the bounds in the single-machine setting. If no gradient noise is present and all the functions $\{F_v\}_{v\in V}$ are the same, then the network terms vanish as there is no difference between X_v^s and $\overline{X}^s$ (recall that the initial conditions are the same for each node, i.e., $X_v^1 = 0$ for all $v \in V$) and optimal tuning of the step sizes recovers the same rates as for serial SGD: $O(1/\sqrt{t})$ for the Lipschitz case and $O(1/t)$ for the smooth case.*

As the matrix P is doubly stochastic, the network average $\overline{X}^s$ admits the following simple evolution:

$$\overline{X}^{s+1} = \overline{X}^s - \eta\frac{1}{n}\sum_{v\in V} G_v^{s+1}. \tag{16}$$

In particular, note that by rearranging the previous expression we get

$$\frac{1}{n}\sum_{v\in V} G_v^{s+1} = \frac{1}{\eta}(\overline{X}^s - \overline{X}^{s+1}), \tag{17}$$

which will be used in the proofs in the next sections.

Before moving on to the next sections and presenting the Optimisation Error bounds, we establish bounds on the network terms that hold in the setting introduced so far. The next proposition bounds the deviation of X_v^s from $\overline{X}^s$ as a function of the second largest eigenvalue in magnitude of the matrix P, i.e., $\sigma_2(P)$. We present different bounds, that either depend on the Lipschitz parameter L or on a *Gradient Noise and Function Deviation Term* κ, as defined in (18). If no gradient noise is present and all the functions $\{F_v\}_{v\in V}$ are the same, then $\kappa = 0$, reflecting the comment in Remark 22.

Proposition 23 (Network term) *Consider the assumptions of Section B.1. Let κ^2 be such that, for any $v \in V, s \geq 1$,*

$$\underbrace{\mathbf{E}\Big[\Big\|G_v^{s+1} - \frac{1}{n}\sum_{\ell=1}^{n}\nabla F_\ell(X_\ell^s)\Big\|^2\Big]}_{\textit{Gradient Noise and Function Deviation Term}} \leq \kappa^2. \tag{18}$$

For any $v \in V, s \geq 1$, we have

$$\mathbf{E}[\|X_v^s - \overline{X}^s\|^2] \leq \eta^2 \min\{L^2, \kappa^2\}\left(2\frac{\log(s\sqrt{n})}{1-\sigma_2(P)} + 1\right)^2.$$

Proof Fix $v \in V, s \geq 1$. By unraveling the updates in (13) and (16), using that $X_v^1 = 0$ for all $v \in V$, we get

$$X_v^s = -\eta\sum_{k=1}^{s-1}\sum_{w\in V} P_{vw}^k G_w^{s-k+1}, \qquad \overline{X}^s = -\eta\sum_{k=1}^{s-1}\sum_{w\in V}(\tfrac{1}{n}11^\top)_{vw} G_w^{s-k+1},$$

where $1 \in \mathbb{R}^n$ is the all-one vector. Using that the rows of the matrix P sum to one, note that for any collection of vectors $\{\nu^k\}_{k=1}^{s-1}$ in $\mathbb{R}^d$ we have

$$X_v^s - \overline{X}^s = \eta \sum_{k=1}^{s-1} \sum_{w \in V} (P^k - \tfrac{1}{n} 1 1^\top)_{vw} (G_w^{s-k+1} - \nu^{s-k}).$$

We have

$$\mathbf{E}[\|X_v^s - \overline{X}^s\|^2] = \mathbf{E}\langle X_v^s - \overline{X}^s, X_v^s - \overline{X}^s \rangle$$
$$\leq \eta^2 \sum_{k,k'=1}^{s-1} \sum_{w,w' \in V} |(P^k - \tfrac{1}{n} 1 1^\top)_{vw}| |(P^{k'} - \tfrac{1}{n} 1 1^\top)_{vw'}| \, \mathbf{E}|\langle G_w^{s-k+1} - \nu^{s-k}, G_{w'}^{s-k'+1} - \nu^{s-k'} \rangle|.$$

By Cauchy-Schwarz's inequality and Hölder's inequality,

$$\mathbf{E}|\langle G_w^{s-k+1} - \nu^{s-k}, G_{w'}^{s-k'+1} - \nu^{s-k'} \rangle| \leq \sqrt{\mathbf{E}[\|G_w^{s-k+1} - \nu^{s-k}\|^2]} \sqrt{\mathbf{E}[\|G_{w'}^{s-k'+1} - \nu^{s-k'}\|^2]},$$

and the above yields

$$\mathbf{E}[\|X_v^s - \overline{X}^s\|^2] \leq \left(\eta \sum_{k=1}^{s-1} \sum_{w \in V} |(P^k - \tfrac{1}{n} 1 1^\top)_{vw}| \sqrt{\mathbf{E}[\|G_w^{s-k+1} - \nu^{s-k}\|^2]} \right)^2.$$

By choosing $\nu^k = 0$ and using (15), we get

$$\mathbf{E}[\|X_v^s - \overline{X}^s\|^2] \leq \eta^2 L^2 \left(\sum_{k=1}^{s-1} \sum_{w \in V} |(P^k - \tfrac{1}{n} 1 1^\top)_{vw}| \right)^2.$$

By choosing $\nu^k = \frac{1}{n} \sum_{\ell=1}^n \nabla F_\ell(X_\ell^k)$ and using the assumption of the proposition, we get

$$\mathbf{E}[\|X_v^s - \overline{X}^s\|^2] \leq \eta^2 \kappa^2 \left(\sum_{k=1}^{s-1} \sum_{w \in V} |(P^k - \tfrac{1}{n} 1 1^\top)_{vw}| \right)^2.$$

Note that $\sum_{k=1}^{s-1} \sum_{w \in V} |(P^k - \frac{1}{n} 1 1^\top)_{vw}| = \sum_{k=1}^{s-1} \|e_v^\top P^k - \frac{1}{n} 1^\top\|_1$, where $e_v \in \mathbb{R}^n$ is the vector with 1 in the coordinate aligning with node v and 0 everywhere else, and $\|\cdot\|_1$ denotes the ℓ_1 norm. Standard results from Perron-Frobenius theory yield, for any $k \geq 1$,

$$\|e_v^\top P^k - \tfrac{1}{n} 1^\top\|_1 \leq \sqrt{n} \|e_v^\top P^k - \tfrac{1}{n} 1^\top\| \leq \sqrt{n} \sigma_2(P)^k.$$

To bound the quantity $\sum_{k=1}^{s-1} \|e_v^\top P^k - \frac{1}{n} 1^\top\|_1$, break the sum and bound each part separately. For the first half use $\|e_v^\top P^k - \frac{1}{n} 1^\top\|_1 \leq \|e_v^\top P^k\|_1 + \|\frac{1}{n} 1^\top\|_1 = 2$ so

$$\sum_{k=1}^{s-1} \|e_v^\top P^k - \tfrac{1}{n} 1^\top\|_1 = \sum_{k=1}^{\tilde{s}} \|e_v^\top P^k - \tfrac{1}{n} 1^\top\|_1 + \sum_{k=\tilde{s}+1}^{s-1} \|e_v^\top P^k - \tfrac{1}{n} 1^\top\|_1 \leq 2\tilde{s} + \sqrt{n} \sum_{k=\tilde{s}+1}^{s-1} \sigma_2(P)^k.$$

Requiring $\sigma_2(P)^k \leq \frac{1}{s\sqrt{n}}$ for k between $\tilde{s}+1$ and $s-1$, set $\tilde{s} = \lfloor \frac{\log(s\sqrt{n})}{\log(\sigma_2(P)^{-1})} \rfloor$. As there are no more than s terms in the sum, using that $\log(x^{-1}) \geq 1 - x$, we get

$$\sum_{k=1}^{s-1} \|e_v^\top P^k - \tfrac{1}{n} 1^\top\|_1 \leq 2\tilde{s} + 1 \leq 2 \frac{\log(s\sqrt{n})}{1 - \sigma_2(P)} + 1.$$

■

B.2. Convex and Lipschitz

The following result controls the evolution of algorithm (13) in the setting defined in Section B.1, under the additional assumption of Lipschitz continuity. The proof is inspired from the analysis in Duchi et al. (2012),

Theorem 24 (Optimisation bound for convex and Lipschitz objectives) *Consider the setting of Section B.1. Let the functions $\{F_v\}_{v\in V}$ be L-Lipschitz. Then, Distributed SGD yields, for any $v \in V$ and $t \geq 1$,*

$$\mathbf{E}\Big[\overline{F}\Big(\frac{1}{t}\sum_{s=1}^{t} X_v^s\Big) - \overline{F}(X^\star)\Big] \leq \frac{1}{t}\sum_{s=1}^{t}\mathbf{E}[\overline{F}(X_v^s) - \overline{F}(X^\star)]$$
$$\leq \underbrace{\frac{3L}{t}\max_{w\in V}\sum_{s=1}^{t}\mathbf{E}\|X_w^s - \overline{X}^s\|}_{\textit{Network Term}} + \underbrace{\frac{1}{t}\sum_{s=1}^{t}\frac{1}{n}\sum_{w\in V}\mathbf{E}\langle G_w^{s+1}, \overline{X}^s - X^\star\rangle}_{\textit{Optimisation Term}}.$$

and the Optimisation Term is upper bounded by $\frac{G^2}{2\eta t} + \frac{\eta L^2}{2}$.

Proof For any $s \geq 1$ and $v \in V$, adding and subtracting the term $\frac{1}{n}\sum_{w\in V} F_w(X_w^s)$, we find

$$\mathbf{E}[\overline{F}(X_v^s) - \overline{F}(X^\star)] = \frac{1}{n}\sum_{w\in V}\mathbf{E}[F_w(X_v^s) - F_w(X_w^s)] + \frac{1}{n}\sum_{w\in V}\mathbf{E}[F_w(X_w^s) - F_w(X^\star)]$$
$$\leq \frac{1}{n}\sum_{w\in V} L\mathbf{E}\|X_v^s - X_w^s\| + \frac{1}{n}\sum_{w\in V}\mathbf{E}\langle G_w^{s+1}, X_w^s - X^\star\rangle,$$

where for the first summand we used the Lipschitz property, and for the second summand we used convexity, assumption (14), and that both $\{X_v^s\}_{v\in V}$ and $X^\star$ are measurable with respect to $\mathcal{F}_s$. In fact, for any $w \in V$, we have

$$F_w(X_w^s) - F_w(X^\star) \leq \langle \partial F_w(X_w^s), X_w^s - X^\star\rangle = \langle \mathbf{E}[G_w^{s+1}|\mathcal{F}_s], X_w^s - X^\star\rangle = \mathbf{E}[\langle G_w^{s+1}, X_w^s - X^\star\rangle|\mathcal{F}_s],$$

so that $\mathbf{E}[F_w(X_w^s) - F_w(X^\star)] \leq \mathbf{E}\langle G_w^{s+1}, X_w^s - X^\star\rangle$ by the tower property of conditional expectations. By adding and subtracting $\overline{X}^s$ and applying the Cauchy-Schwarz's inequality, we have

$$\mathbf{E}\langle G_w^{s+1}, X_w^s - X^\star\rangle \leq \mathbf{E}[\|G_w^{s+1}\|\|X_w^s - \overline{X}^s\|] + \mathbf{E}\langle G_w^{s+1}, \overline{X}^s - X^\star\rangle,$$

and the first term on the right-hand side is further bounded by using Jensen's inequality and the fact that $(X_w^s - \overline{X}^s)$ is $\mathcal{F}_s$-measurable, along with assumption (15), giving

$$\mathbf{E}[\|G_w^{s+1}\|\|X_w^s - \overline{X}^s\|] \leq \mathbf{E}[(\mathbf{E}[\|G_w^{s+1}\|^2|\mathcal{F}_s])^{1/2}\|X_w^s - \overline{X}^s\|] \leq L\mathbf{E}\|X_w^s - \overline{X}^s\|.$$

All together with $\|X_v^t - X_w^s\| \leq 2\max_{w'\in V}\|X_{w'}^s - \overline{X}^s\|$ we have

$$\frac{1}{t}\sum_{s=1}^{t}\mathbf{E}[\overline{F}(X_v^s) - \overline{F}(X^\star)] \leq \frac{3L}{t}\max_{w\in V}\sum_{s=1}^{t}\mathbf{E}\|X_w^s - \overline{X}^s\| + \frac{1}{t}\sum_{s=1}^{t}\frac{1}{n}\sum_{w\in V}\mathbf{E}\langle G_w^{s+1}, \overline{X}^s - X^\star\rangle.$$

To bound the Optimisation Term we proceed as follows. Using (17) and that $\langle a, b\rangle = (\|a\|^2 + \|b\|^2 - \|a-b\|^2)/2$ we obtain

$$\begin{aligned}\frac{1}{n}\sum_{w\in V}\mathbf{E}\langle G_w^{s+1}, \overline{X}^s - X^\star\rangle &= \frac{1}{\eta}\mathbf{E}\langle \overline{X}^s - \overline{X}^{s+1}, \overline{X}^s - X^\star\rangle \\ &= \frac{1}{2\eta}(\mathbf{E}[\|\overline{X}^{s+1} - \overline{X}^s\|^2] + \mathbf{E}[\|\overline{X}^s - X^\star\|^2] - \mathbf{E}[\|\overline{X}^{s+1} - X^\star\|^2]) \\ &\leq \frac{1}{2\eta}\Big(\mathbf{E}[\|\overline{X}^s - X^\star\|^2] - \mathbf{E}[\|\overline{X}^{s+1} - X^\star\|^2] + \eta^2\mathbf{E}\Big[\Big\|\frac{1}{n}\sum_{w\in V} G_w^{s+1}\Big\|^2\Big]\Big) \\ &\leq \frac{1}{2\eta}(\mathbf{E}[\|\overline{X}^s - X^\star\|^2] - \mathbf{E}[\|\overline{X}^{s+1} - X^\star\|^2] + \eta^2 L^2),\end{aligned}$$

where we used Cauchy-Schwarz's and Hölder's inequalities, along with assumption (15), to get

$$\begin{aligned}\mathbf{E}\Big[\Big\|\frac{1}{n}\sum_{w\in V} G_w^{s+1}\Big\|^2\Big] = \frac{1}{n^2}\sum_{u,w\in V}\mathbf{E}\langle G_u^{s+1}, G_w^{s+1}\rangle &\leq \frac{1}{n^2}\sum_{u,w\in V}\mathbf{E}[\|G_u^{s+1}\|\|G_w^{s+1}\|] \\ &\leq \frac{1}{n^2}\sum_{u,w\in V}\sqrt{\mathbf{E}[\|G_u^{s+1}\|^2]}\sqrt{\mathbf{E}[\|G_w^{s+1}\|^2]} \leq L^2. \qquad (19)\end{aligned}$$

Summing over s, using that $X_v^1 = 0$ for all $v \in V$ and that $\mathbf{E}[\|X^\star\|^2] \leq G^2$, we get the following bound for the Optimisation Term

$$\frac{1}{t}\sum_{s=1}^{t}\frac{1}{n}\sum_{w\in V}\mathbf{E}\langle G_w^{s+1}, \overline{X}^s - X^\star\rangle \leq \frac{1}{2\eta t}\mathbf{E}[\|\overline{X}^1 - X^\star\|^2] + \frac{\eta L^2}{2} \leq \frac{G^2}{2\eta t} + \frac{\eta L^2}{2}.$$

■

Corollary 25 *Consider the assumptions of Section B.1. Let the functions $\{F_v\}_{v\in V}$ be L-Lipschitz. Then, Distributed SGD yields, for any $v \in V$ and $t \geq 1$,*

$$\mathbf{E}\Big[\overline{F}\Big(\frac{1}{t}\sum_{s=1}^{t} X_v^s\Big) - \overline{F}(X^\star)\Big] \leq \frac{1}{t}\sum_{s=1}^{t}\mathbf{E}[\overline{F}(X_v^s) - \overline{F}(X^\star)] \leq \frac{\eta L^2}{2}\Big(19\frac{\log(t\sqrt{n})}{1-\sigma_2(P)}\Big) + \frac{G^2}{2\eta t}.$$

Proof It follows from Theorem 24 and Proposition 23, as $\mathbf{E}\|X_v^s - \overline{X}^s\| \leq \sqrt{\mathbf{E}[\|X_v^s - \overline{X}^s\|^2]}$ by Jensen's inequality. ■

B.3. Convex and Smooth

The following result controls the evolution of algorithm (13) in the setting defined in Section B.1, under the additional assumption of smoothness. The proof is inspired by the one given Dekel et al. (2012) for single-machine serial SGD applied to smooth losses, the specific exposition of which more closely follows Bubeck et al. (2015). The bound that we give is made of three components: the Optimisation Term that decays like $1/t$; the Gradient Noise Term that captures the average noise of the gradient across the graph; the Network Term that controls the deviation of the algorithm from its network average.

Theorem 26 (Optimisation bound for convex and smooth objectives) *Consider the Assumptions of Section B.1. Let the functions $\{F_v\}_{v\in V}$ be β-smooth. Then, Distributed SGD with $\eta = 1/(\beta + 1/\rho)$ and $\rho \geq 0$, yields, for any $v \in V$ and $t \geq 1$,*

$$\mathbf{E}\Big[\overline{F}\Big(\frac{1}{t}\sum_{s=1}^{t} X_v^{s+1}\Big) - \overline{F}(X^\star)\Big] \leq \frac{1}{t}\sum_{s=1}^{t}\mathbf{E}[\overline{F}(X_v^{s+1}) - \overline{F}(X^\star)]$$

$$\leq \underbrace{\frac{1}{t}\sum_{s=1}^{t}\Big(L\mathbf{E}\|X_v^{s+1} - \overline{X}^{s+1}\| + \beta \max_{w\in V}\mathbf{E}[\|X_w^{s+1} - \overline{X}^{s+1}\|^2] + \frac{\beta}{2}\Big(1+\beta\rho\Big)\max_{w\in V}\mathbf{E}[\|X_w^{s} - \overline{X}^{s}\|^2]\Big)}_{\textit{Network Term}}$$

$$+ \underbrace{\frac{\rho}{2}\frac{1}{t}\sum_{s=1}^{t}\mathbf{E}\Big[\Big\|\frac{1}{n}\sum_{w\in V}(G_w^{s+1} - \nabla F_w(X_w^s))\Big\|^2\Big]}_{\textit{Gradient Noise Term}}$$

$$+ \underbrace{\frac{1}{t}\sum_{s=1}^{t}\Big(\frac{1}{n}\sum_{w\in V}\mathbf{E}\langle G_w^{s+1}, \overline{X}^{s+1} - X^\star\rangle + \frac{1}{2}\Big(\beta + \frac{1}{\rho}\Big)\mathbf{E}[\|\overline{X}^{s+1} - \overline{X}^{s}\|^2]\Big)}_{\textit{Optimisation Term}},$$

and the Optimisation Term is upper bounded by $\frac{1}{2}(\beta + \frac{1}{\rho})\frac{G^2}{t}$.

Proof Recall that if a function $f : \mathbb{R}^d \to \mathbb{R}$ is β-smooth then for any $x, y \in \mathbb{R}^d$ we have (Nesterov, 2013) $f(x) - f(y) \leq \langle \nabla f(y), x - y\rangle + \frac{\beta}{2}\|x - y\|^2$. Fix $s \geq 1$, $v \in V$. Consider the following decomposition.

$$\overline{F}(X_v^{s+1}) - \overline{F}(X^\star) = \underbrace{\overline{F}(X_v^{s+1}) - \overline{F}(\overline{X}^{s+1})}_{\text{Term (a)}} + \underbrace{\overline{F}(\overline{X}^{s+1}) - \overline{F}(X^\star)}_{\text{Term (b)}}. \tag{20}$$

Term (a). To bound Term (a), we use smoothness and convexity to get

$$\overline{F}(X_v^{s+1}) - \overline{F}(\overline{X}^{s+1}) = \frac{1}{n}\sum_{w\in V}\Big(F_w(X_v^{s+1}) - F_w(X_w^{s+1}) + F_w(X_w^{s+1}) - F_w(\overline{X}^{s+1})\Big)$$

$$\leq \frac{1}{n}\sum_{w\in V}\Big(\langle \nabla F_w(X_w^{s+1}), X_v^{s+1} - X_w^{s+1}\rangle + \frac{\beta}{2}\|X_v^{s+1} - X_w^{s+1}\|^2 + \langle \nabla F_w(X_w^{s+1}), X_w^{s+1} - \overline{X}^{s+1}\rangle\Big)$$

$$= \frac{1}{n}\sum_{w\in V}\Big(\langle \nabla F_w(X_w^{s+1}), X_v^{s+1} - \overline{X}^{s+1}\rangle + \frac{\beta}{2}\|X_v^{s+1} - X_w^{s+1}\|^2\Big).$$

As $\nabla F_w(X_w^{s+1}) = \mathbf{E}[G_w^{s+2}|\mathcal{F}_{s+1}]$ and $\{X_w^{s+1}\}_{w\in V}$ are $\mathcal{F}_{s+1}$-measurable, we get

$$\begin{aligned}\langle \nabla F_w(X_w^{s+1}), X_v^{s+1} - \overline{X}^{s+1}\rangle &= \mathbf{E}[\langle G_w^{s+2}, X_v^{s+1} - \overline{X}^{s+1}\rangle|\mathcal{F}_{s+1}] \\ &\leq \mathbf{E}[\|G_w^{s+2}\|\|X_v^{s+1} - \overline{X}^{s+1}\||\mathcal{F}_{s+1}] \\ &\leq \sqrt{\mathbf{E}[\|G_w^{s+2}\|^2|\mathcal{F}_{s+1}]}\|X_v^{s+1} - \overline{X}^{s+1}\| \\ &\leq L\|X_v^{s+1} - \overline{X}^{s+1}\|,\end{aligned}$$

where we used Cauchy-Schwarz's inequality, Jensen's inequality, and $\mathbf{E}[\|G_w^{s+2}\|^2|\mathcal{F}_{s+1}] \leq L^2$. Thus,

$$\mathbf{E}[\overline{F}(X_v^{s+1}) - \overline{F}(\overline{X}^{s+1})] \leq L\mathbf{E}\|X_v^{s+1} - \overline{X}^{s+1}\| + \beta \max_{w\in V} \mathbf{E}[\|X_w^{s+1} - \overline{X}^{s+1}\|^2]. \tag{21}$$

Term (b). To bound Term (b), we use smoothness to find a bound that involves a telescoping sum whose terms cancel out when we take the summation over time s. Using smoothness, adding and subtracting $\langle G_w^{s+1}, \overline{X}^{s+1}, \overline{X}^s\rangle = \langle G_w^{s+1}, \overline{X}^{s+1} - X^\star\rangle + \langle G_w^{s+1}, X^\star - \overline{X}^{s+1}\rangle$ and using Cauchy-Schwarz's inequality ($2\langle a, b\rangle \leq \rho\|a\|^2 + \|b\|^2/\rho$ for $\rho \geq 0$) we get

$$\begin{aligned}
\overline{F}(\overline{X}^{s+1}) - \overline{F}(\overline{X}^s) &\leq \frac{1}{n}\sum_{w\in V}\langle \nabla F_w(\overline{X}^s), \overline{X}^{s+1} - \overline{X}^s\rangle + \frac{\beta}{2}\|\overline{X}^{s+1} - \overline{X}^s\|^2 \\
&= \Big\langle \frac{1}{n}\sum_{w\in V}(\nabla F_w(\overline{X}^s) - G_w^{s+1}), \overline{X}^{s+1} - \overline{X}^s\Big\rangle + \frac{1}{n}\sum_{w\in V}\langle G_w^{s+1}, \overline{X}^{s+1} - X^\star\rangle \\
&\quad + \frac{1}{n}\sum_{w\in V}\langle G_w^{s+1}, X^\star - \overline{X}^s\rangle + \frac{\beta}{2}\|\overline{X}^{s+1} - \overline{X}^s\|^2 \\
&\leq \frac{\rho}{2}\Big\|\frac{1}{n}\sum_{w\in V}(\nabla F_w(\overline{X}^s) - G_w^{s+1})\Big\|^2 + \frac{1}{n}\sum_{w\in V}\langle G_w^{s+1}, \overline{X}^{s+1} - X^\star\rangle \\
&\quad + \frac{1}{n}\sum_{w\in V}\langle G_w^{s+1}, X^\star - \overline{X}^s\rangle + \frac{1}{2}\Big(\beta + \frac{1}{\rho}\Big)\|\overline{X}^{s+1} - \overline{X}^s\|^2.
\end{aligned} \tag{22}$$

Adding $\overline{F}(\overline{X}^s)$ to both sides, taking expectation, using that $\{X_w^s\}_{w\in V}$ and $X^\star$ are $\mathcal{F}_s$-measurable, and that $\mathbf{E}\langle[G_w^{s+1}, X^\star - \overline{X}^s\rangle|\mathcal{F}_s] = \langle \nabla F_w(X_w^s), X^\star - \overline{X}^s\rangle$, we get

$$\begin{aligned}
\mathbf{E}[\overline{F}(\overline{X}^{s+1}) - \overline{F}(X^\star)] &\leq \mathbf{E}[\overline{F}(\overline{X}^s) - \overline{F}(X^\star)] + \frac{\rho}{2}\mathbf{E}\Big[\Big\|\frac{1}{n}\sum_{w\in V}(\nabla F_w(\overline{X}^s) - G_w^{s+1})\Big\|^2\Big] \\
&\quad + \frac{1}{n}\sum_{w\in V}\mathbf{E}\langle G_w^{s+1}, \overline{X}^{s+1} - X^\star\rangle + \frac{1}{2}\Big(\beta + \frac{1}{\rho}\Big)\mathbf{E}[\|\overline{X}^{s+1} - \overline{X}^s\|^2] \\
&\quad + \frac{1}{n}\sum_{w\in V}\mathbf{E}\langle \nabla F_w(X_w^s), X^\star - \overline{X}^s\rangle.
\end{aligned} \tag{23}$$

To bound the first term on the right-hand side of bound (23) and cancel the dependence on $X^\star$ from the term $\langle \nabla F_w(X_w^s), X^\star - \overline{X}^s\rangle$, note that by convexity and smoothness we get

$$\begin{aligned}
\mathbf{E}[\overline{F}(\overline{X}^s) - \overline{F}(X^\star)] &= \frac{1}{n}\sum_{w\in V}\mathbf{E}[F_w(\overline{X}^s) - F_w(X_w^s) + F_w(X_w^s) - F_w(X^\star)] \\
&= \frac{1}{n}\sum_{w\in V}\mathbf{E}[F_w(\overline{X}^s) - F_w(X_w^s) + \langle \nabla F_w(X_w^s), X_w^s - \overline{X}^s\rangle + \langle \nabla F_w(X_w^s), \overline{X}^s - X^\star\rangle] \\
&\leq \frac{\beta}{2}\max_{w\in V}\mathbf{E}\|X_w^s - \overline{X}^s\|^2 + \frac{1}{n}\sum_{w\in V}\mathbf{E}\langle \nabla F_w(X_w^s), \overline{X}^s - X^\star\rangle.
\end{aligned} \tag{24}$$

To bound the second term on the right-hand side of bound (23), note that

$$\mathbf{E}\Big[\Big\|\frac{1}{n}\sum_{w\in V}(\nabla F_w(\overline{X}^s)-G_w^{s+1})\Big\|^2\Big]=\mathbf{E}\Big[\Big\|\frac{1}{n}\sum_{w\in V}\Big(\nabla F_w(\overline{X}^s)-\nabla F_w(X_w^s)+\nabla F_w(X_w^s)-G_w^{s+1}\Big)\Big\|^2\Big]$$
$$=\mathbf{E}\Big[\Big\|\frac{1}{n}\sum_{w\in V}(\nabla F_w(\overline{X}^s)-\nabla F_w(X_w^s))\Big\|^2\Big]+\mathbf{E}\Big[\Big\|\frac{1}{n}\sum_{w\in V}(\nabla F_w(X_w^s)-G_w^{s+1})\Big\|^2\Big], \tag{25}$$

where we used that the cross terms are zero as $\mathbf{E}[G_w^{s+1}|\mathcal{F}_s]=\nabla F_w(X_w^s)$ and both $\{F_w\}_{w\in V}$ and $\{X_w^s\}_{w\in V}$ are $\mathcal{F}_s$-measurable. The first term in (25) can be bounded as follows:

$$\begin{aligned}
&\mathbf{E}\Big[\Big\|\frac{1}{n}\sum_{w\in V}(\nabla F_w(X_w^s)-\nabla F_w(\overline{X}^s))\Big\|^2\Big]\\
&=\frac{1}{n^2}\sum_{w,l\in V}\mathbf{E}\langle\nabla F_w(X_w^s)-\nabla F_w(\overline{X}^s),\nabla F_l(X_l^s)-\nabla F_l(\overline{X}^s)\rangle\\
&\le\frac{1}{n^2}\sum_{w,l\in V}\mathbf{E}\big[\|\nabla F_w(X_w^s)-\nabla F_w(\overline{X}^s)\|\|\nabla F_l(X_l^s)-\nabla F_l(\overline{X}^s)\|\big]\\
&\le\frac{\beta^2}{n^2}\sum_{w,l\in V}\mathbf{E}\big[\|X_w^s-\overline{X}^s\|\|X_l^s-\overline{X}^s\|\big]\\
&\le\frac{\beta^2}{n^2}\sum_{w,l\in V}\sqrt{\mathbf{E}\big[\|X_w^s-\overline{X}^s\|^2\big]}\sqrt{\mathbf{E}\big[\|X_l^s-\overline{X}^s\|^2\big]}\\
&\le\beta^2\max_{w\in V}\mathbf{E}\big[\|X_w^s-\overline{X}^s\|^2\big],
\end{aligned}\tag{26}$$

where applied Cauchy-Schwarz's inequality, smoothness, and Hölder's inequality. Plugging (24), (25), and (26) into (23) we get the following bound for the expected value of term (b):

$$\mathbf{E}[\overline{F}(\overline{X}^{s+1})-\overline{F}(X^\star)]\le\frac{\beta}{2}\Big(1+\beta\rho\Big)\max_{w\in V}\mathbf{E}[\|X_w^s-\overline{X}^s\|^2]+\frac{\rho}{2}\mathbf{E}\Big[\Big\|\frac{1}{n}\sum_{w\in V}(\nabla F_w(X_w^s)-G_w^{s+1})\Big\|^2\Big]$$
$$+\frac{1}{n}\sum_{w\in V}\mathbf{E}\langle G_w^{s+1},\overline{X}^{s+1}-X^\star\rangle+\frac{1}{2}\Big(\beta+\frac{1}{\rho}\Big)\mathbf{E}[\|\overline{X}^{s+1}-\overline{X}^s\|^2]. \tag{27}$$

Term (a) + Term (b). The main result in the theorem follows by using bounds (21) and (27) to bound Term (a) and Term (b) in (20), taking the summation over time from $s=1$ to $s=t$.

To bound the Optimisation Term, use (17) and that $2\langle a,b\rangle=\|a\|^2+\|b\|^2-\|a-b\|^2$ so that

$$\begin{aligned}
\frac{1}{n}\sum_{w\in V}\langle G_w^{s+1},\overline{X}^{s+1}-X^\star\rangle&=\frac{1}{\eta}\langle\overline{X}^s-\overline{X}^{s+1},\overline{X}^{s+1}-X^\star\rangle\\
&=-\frac{1}{\eta}\langle\overline{X}^{s+1}-\overline{X}^s,\overline{X}^{s+1}-X^\star\rangle\\
&=\frac{1}{2\eta}\Big(-\|\overline{X}^{s+1}-\overline{X}^s\|^2-\|\overline{X}^{s+1}-X^\star\|^2+\|\overline{X}^s-X^\star\|^2\Big).
\end{aligned}$$

The choice $\eta=\frac{1}{\beta+1/\rho}$ leads to the cancellation of the quantity $\|\overline{X}^{s+1}-\overline{X}^s\|^2$ in the Optimisation Term. The telescoping sum over time, using that $X_w^1=0$ for all $w\in V$ and the assumption

$\mathbf{E}[\|X^\star\|^2] \leq G^2$, yields the final result. ■

As for single-machine serial SGD (Dekel et al., 2012), the error bound that we give in Theorem 26 for the smooth case exhibits explicit dependence on the gradient noise, which in our setting is averaged out across the network. As far as the following corollary is concerned, we assume a time-uniform control on the gradient noise, namely,

$$\mathbf{E}\Big[\Big\|\frac{1}{n}\sum_{w\in V}(G_w^{s+1}-\nabla F_w(X_w^s))\Big\|^2\Big] \leq \sigma^2 \tag{28}$$

for any $s \geq 1$.

Corollary 27 *Consider the Assumptions of Section B.1. Let the functions $\{F_v\}_{v\in V}$ be β-smooth and satisfy both* (18) *and* (28)*. Then, Distributed SGD with $\eta = 1/(\beta + 1/\rho)$ and $\rho \geq 0$, yields, for any $v \in V$ and $t \geq 1$,*

$$\mathbf{E}\Big[\overline{F}\Big(\frac{1}{t}\sum_{s=1}^{t} X_v^{s+1}\Big) - \overline{F}(X^\star)\Big] \leq \frac{1}{t}\sum_{s=1}^{t}\mathbf{E}[\overline{F}(X_v^{s+1}) - \overline{F}(X^\star)]$$
$$\leq \frac{\rho}{2}\sigma^2 + \frac{(\beta+1/\rho)G^2}{2t} + \frac{3\kappa}{\beta+1/\rho}\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)}\Big(L + \frac{3}{2}\frac{\beta(3+\beta\rho)\kappa}{\beta+1/\rho}\frac{\log((t+1)\sqrt{n})}{1-\sigma_2(P)}\Big)$$

Proof It follows from Theorem 26 and Proposition 23. ■

B.4. Assumptions for Distributed SGD (1)

This section verifies that the more general assumptions considered in this Appendix for Distributed SGD (13) are satisfied within the context of the main body of this work, that is, for Distributed SGD (1) as described within Section 3. This is performed by placing Distributed SGD (1) into the context Distributed SGD (13) as follows. Let the random objective functions be $F_v(x) = R_v(x) = \frac{1}{m}\sum_{k=1}^{m}\ell(x, Z_{v,k})$ for $v \in V$, which yields the network average $\overline{F}(x) = R(x)$. Consider the following stochastic gradients, for $v \in V$ and $s \geq 1$,

$$G_v^{s+1} = \partial\ell(X_v^s, Z_{v,K_v^{s+1}}),$$

where K_v^s is a uniform random variable on $[m]$. Let $\mathcal{F}_1$ be the σ-algebra generated by the data sets $\mathcal{D}$. For any $s \geq 2$, let $\mathcal{F}_s$ contain the σ-algebra generated by the data sets $\mathcal{D}$ and the uniform random variables up to time s $\{K_v^2, \ldots, K_v^s\}_{v\in V}$. The random functions $\{F_v\}_{v\in V}$ and their optimal value $X^\star$ are $\mathcal{F}_s$-measurable, as $\mathcal{F}_s$ contains the σ-algebra generated by $\mathcal{D}$. The iterates $\{X_v^k\}_{k\leq s, v\in V}$ are also $\mathcal{F}_s$-measurable, as $\mathcal{F}_s$ contains the σ-algebra generated by $\{K_v^2, \ldots, K_v^s\}_{v\in V}$. We now check that assumption (14) and assumption (15) are satisfied. The following hold for any $s \geq 1$.

- Assumption (14) on the unbiasedness of the subgradient estimators is satisfied as for any $v \in V$ we have

$$\mathbf{E}[G_v^{s+1}|\mathcal{F}_s] = \mathbf{E}[\partial\ell(X_v^s, Z_{v,K_v^{s+1}})\big|\mathcal{F}_s] = \frac{1}{m}\sum_{k=1}^{m}\partial\ell(X_v^s, Z_{v,k}) \in \partial F_v(X_v^s),$$

  where have used that the sum of subgradients belong to the subgradient of sums.

- Assumption (15) on the boundedness of the second moment of the subgradients is satisfied as for any $v \in V$ we have

$$\mathbf{E}[\|G_v^{s+1}\|^2|\mathcal{F}_s] = \mathbf{E}[\|\partial\ell(X_v^s, Z_{v,K_v^{s+1}})\|^2\big|\mathcal{F}_s] = \frac{1}{m}\sum_{k=1}^{m}\|\partial\ell(X_v^s, Z_{v,k})\|^2 \leq L^2,$$

  where we have used that the function $\ell(\,\cdot\,, z)$ is L-Lipschitz for all $z \in Z$.

References

Alekh Agarwal and John C. Duchi. Distributed Delayed Stochastic Optimization. In *Advances in Neural Information Processing Systems*, pages 873–881, 2011.

Avleen S. Bijral, Anand D. Sarwate, and Nathan Srebro. Data-Dependent Convergence for Consensus Stochastic Optimization. *IEEE Transactions on Automatic Control*, 62(9):4483–4498, 2017.

Olivier Bousquet and Léon Bottou. The Tradeoffs of Large Scale Learning. In *Advances in Neural Information Processing Systems*, pages 161–168, 2008.

Olivier Bousquet and André Elisseeff. Stability and Generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011.

Sébastien Bubeck et al. Convex Optimization: Algorithms and Complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal Distributed Online Prediction using Mini-batches. *Journal of Machine Learning Research*, 13(Jan):165–202, 2012.

Aymeric Dieuleveut and Francis Bach. Nonparametric Stochastic Approximation with Large Stepsizes. *The Annals of Statistics*, 44(4):1363–1399, 2016.

Alexandros G. Dimakis, Soummya Kar, José M.F. Moura, Michael G. Rabbat, and Anna Scaglione. Gossip Algorithms for Distributed Signal Processing. *Proceedings of the IEEE*, 98(11):1847–1864, 2010.

John C. Duchi, Alekh Agarwal, and Martin J. Wainwright. Dual Averaging for Distributed Optimization: Convergence Analysis and Network Scaling. *IEEE Transactions on Automatic Control*, 57(3):592–606, 2012.

Moritz Hardt, Benjamin Recht, and Yoram Singer. Train Faster, Generalize Better: Stability of Stochastic Gradient Descent. In *International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1225–1234, 2016.

Bjorn Johansson, Maben Rabi, and Mikael Johansson. A Simple Peer-to-Peer Algorithm for Distributed Optimization in Sensor Networks. In *Decision and Control, 2007 46th IEEE Conference on*, pages 4705–4710. IEEE, 2007.

Björn Johansson, Maben Rabi, and Mikael Johansson. A Randomized Incremental Subgradient Method for Distributed Optimization in Networked Systems. *SIAM Journal on Optimization*, 20 (3):1157–1170, 2009.

Michael Kearns and Dana Ron. Algorithmic Stability and Sanity-Check Bounds for Leave-One-Out Cross-Validation. *Neural computation*, 11(6):1427–1453, 1999.

Guanghui Lan. An Optimal Method for Stochastic Composite Optimization. *Mathematical Programming*, 133(1):365–397, 2012.

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can Decentralized Algorithms Outperform Centralized Algorithms? A Case Study for Decentralized Parallel Stochastic Gradient Descent. In *Advances in Neural Information Processing Systems*, pages 5330–5340, 2017.

Junhong Lin and Volkan Cevher. Optimal Distributed Learning with Multi-pass Stochastic Gradient Methods. *Proceedings of the 35th International Conference on Machine Learning*, page 27, 2018.

Junhong Lin and Lorenzo Rosasco. Optimal Rates for Multi-Pass Stochastic Gradient Methods. *Journal of Machine Learning Research*, 18(97):1–47, 2017.

Junhong Lin, Raffaello Camoriano, and Lorenzo Rosasco. Generalization Properties and Implicit Regularization for Multiple Passes SGM. In *International Conference on Machine Learning*, pages 2340–2348, 2016a.

Junhong Lin, Lorenzo Rosasco, and Ding-Xuan Zhou. Iterative Regularization for Learning with Convex Loss Functions. *Journal of Machine Learning Research*, 17(1):2718–2755, 2016b.

Ilan Lobel and Asuman Ozdaglar. Distributed Subgradient Methods for Convex Optimization over Random Networks. *IEEE Transactions on Automatic Control*, 56(6):1291–1306, 2011.

Ion Matei and John S. Baras. Performance Evaluation of the Consensus-Based Distributed Subgradient Method Under Random Communication Topologies. *IEEE Journal of Selected Topics in Signal Processing*, 5(4):754–771, 2011.

Aryan Mokhtari and Alejandro Ribeiro. DSA: Decentralized Double Stochastic Averaging Gradient Algorithm. *Journal of Machine Learning Research*, 17(61):1–35, 2016.

Sayan Mukherjee, Partha Niyogi, Tomaso Poggio, and Ryan Rifkin. Learning Theory: Stability is Sufficient for Generalization and Necessary and Sufficient for Consistency of Empirical Risk Minimization. *Advances in Computational Mathematics*, 25(1-3):161–193, 2006.

Angelia Nedic and Asuman Ozdaglar. Distributed Subgradient Methods for Multi-Agent Optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

Angelia Nedić, Alex Olshevsky, Asuman Ozdaglar, and John N. Tsitsiklis. On Distributed Averaging Algorithms and Quantization Effects. *IEEE Transactions on Automatic Control*, 54(11): 2506–2517, 2009.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87. Springer Science & Business Media, 2013.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Oliver Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Aalexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

S. Sundhar Ram, Angelia Nedic, and Venugopal V. Veeravalli. Distributed Subgradient Projection Algorithm for Convex Optimization. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 3653–3656. IEEE, 2009.

Srinivasan Sundhar Ram, Angelia Nedić, and Venugopal V. Veeravalli. Distributed Stochastic Subgradient Projection Algorithms for Convex Optimization. *Journal of Optimization Theory and Applications*, 147(3):516–545, 2010.

William H. Rogers and Terry J. Wagner. A Finite Sample Distribution-Free Performance Bound for Local Discrimination Rules. *The Annals of Statistics*, pages 506–514, 1978.

Ali H. Sayed. Adaptive Networks. *Proceedings of the IEEE*, 102(4):460–497, 2014.

Devavrat Shah. Gossip algorithms. *Foundations and Trends® in Networking*, 3(1):1–125, 2009.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, Stability and Uniform Convergence. *Journal of Machine Learning Research*, 11(Oct):2635–2670, 2010.

Ohad Shamir and Nathan Srebro. Distributed Stochastic Optimization and Learning. In *Communication, Control, and Computing (Allerton), 2014*, pages 850–857. IEEE, 2014.

Ohad Shamir, Nathan Srebro, and Tong Zhang. Communication-Efficient Distributed Optimization using an Approximate Newton-Type Method. In *International Conference on Machine Learning*, pages 1000–1008, 2014.

Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. Extra: An Exact First-Order Algorithm for Decentralized Consensus Optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

Pierre Tarres and Yuan Yao. Online Learning as Stochastic Approximation of Regularization Paths: Optimality and Almost-Sure Convergence. *IEEE Transactions on Information Theory*, 60(9): 5716–5735, 2014.

John Tsitsiklis, Dimitri Bertsekas, and Michael Athans. Distributed Asynchronous Deterministic and Stochastic Gradient Optimization Algorithms. *IEEE transactions on automatic control*, 31 (9):803–812, 1986.

John Nikolas Tsitsiklis. Problems in Decentralized Decision Making and Computation. Technical report, Massachusetts Inst Of Tech Cambridge Lab For Information And Decision Systems, 1984.

Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, Heidelberg, 1995.

Lin Xiao. Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization. *Journal of Machine Learning Research*, 11(Oct):2543–2596, 2010.

Mu Yang and Choon Yik Tang. Distributed Estimation of Graph Spectrum. In *2015 American Control Conference (ACC)*, pages 2703–2708. IEEE, 2015.

Peng Yang, Randy A Freeman, Geoffrey J Gordon, Kevin M Lynch, Siddhartha S Srinivasa, and Rahul Sukthankar. Decentralized Estimation and Control of Graph Connectivity for Mobile Sensor Networks. *Automatica*, 46(2):390–396, 2010.

Yiming Ying and Massimiliano Pontil. Online Gradient Descent Learning Algorithms. *Foundations of Computational Mathematics*, 8(5):561–596, 2008.

Yuchen Zhang and Xiao Lin. DiSCO: Distributed Optimization for Self-concordant Empirical Loss. In *International Conference on Machine Learning*, pages 362–370, 2015.

Yuchen Zhang, Martin J. Wainwright, and John C. Duchi. Communication-Efficient Algorithms for Statistical Optimization. In *Advances in Neural Information Processing Systems*, pages 1502–1510, 2012.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

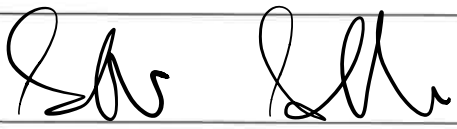
Title of Paper	Graph-Dependent Implicit Regularisation for Distributed Stochastic SubgradientDescent
Publication Status	☐Published
Publication Details	"Graph-Dependent Implicit Regularisation for Distributed Stochas-tic Subgradient Descent", Dominic Richards, Patrick Rebeschini. In Journal of Machine Learning Research, 2020

Student Confirmation

Student Name:	Dominic Richards		
Contribution to the Paper	Derived technical contributions of the manuscript including generalisation and optimistaion error bounds. Provided first bound on network error, which was later refined by Patrick Rebeschini. Wrote first draft of manuscript, with later version written alongside Patrick Rebeschini. Wrote first draft of response to the reviewers and implemented feedback from reviewers.		
Signature		Date	04/01/2021

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	PATRICK REBESCHINI ASSOCIATE PROFESSOR		
Supervisor comments			
Signature		Date	11/01/2021

This completed form should be included in the thesis, at the end of the relevant chapter.

3

Optimal Statistical Rates for Decentralised Non-Parametric Regression with Linear Speed-Up

Optimal Statistical Rates for Decentralised Non-Parametric Regression with Linear Speed-Up

Dominic Richards
Department of Statistics
University of Oxford
24-29 St Giles', Oxford, OX1 3LB
dominic.richards@spc.ox.ac.uk

Patrick Rebeschini
Department of Statistics
University of Oxford
24-29 St Giles', Oxford, OX1 3LB
patrick.rebeschini@stats.ox.ac.uk

Abstract

We analyse the learning performance of Distributed Gradient Descent in the context of multi-agent decentralised non-parametric regression with the square loss function when i.i.d. samples are assigned to agents. We show that if agents hold sufficiently many samples with respect to the network size, then Distributed Gradient Descent achieves optimal statistical rates with a number of iterations that scales, *up to a threshold*, with the inverse of the spectral gap of the gossip matrix divided by the number of samples owned by each agent raised to a problem-dependent power. The presence of the threshold comes from statistics. It encodes the existence of a "big data" regime where the number of required iterations does *not* depend on the network topology. In this regime, Distributed Gradient Descent achieves optimal statistical rates with the *same* order of iterations as gradient descent run with all the samples in the network. Provided the communication delay is sufficiently small, the distributed protocol yields a *linear* speed-up in runtime compared to the single-machine protocol. This is in contrast to decentralised optimisation algorithms that do not exploit statistics and only yield a linear speed-up in graphs where the spectral gap is bounded away from zero. Our results exploit the statistical concentration of quantities held by agents and shed new light on the interplay between statistics and communication in decentralised methods. Bounds are given in the standard non-parametric setting with source/capacity assumptions.

1 Introduction

In machine learning a canonical goal is to use training data sampled independently from an unknown distribution to fit a model that performs well on unseen data from the same distribution. With a loss function measuring the performance of a model on a data point, a common approach is to find a model that minimises the average loss on the training data with some form of *explicit* regularisation to control model complexity and avoid overfitting. Due to the increasingly large size of datasets and high model complexity, direct minimisation of the regularised problem is posing more and more computational challenges. This has led to growing interest in approaches that improve models incrementally using gradient descent methods [8], where model complexity is controlled through forms of *implicit/algorithmic* regularisation such as early stopping and step-size tuning [57, 58, 27].

The growth in the size of modern datasets has also meant that the coordination of multiple machines is often required to fit machine learning models. In the centralised server-clients setup, a single machine (server) is responsible to aggregate and disseminate information to other machines (clients) in what is an effective star topology. In some settings, such as ad-hoc wireless and peer-to-peer networks, network instability, bandwidth limitation and privacy concerns make centralised approaches less feasible. This has motivated research into scalable methods that can avoid the bottleneck

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

and vulnerability introduced by the presence of a central authority. Such solutions are called "decentralised", as no single entity is responsible for the collection and dissemination of information: machines communicate with neighbours in a network structure that encodes communication channels.

Since the early works [52, 53] to the more recent work [22, 34, 33, 23, 29, 30, 10, 18, 47, 31], problems in decentralised multi-agent optimisation have often been treated as a particular instance of consensus optimisation. In this framework, a network of machines or agents collaborate to minimise the average of functions held by individual agents, hence "reaching consensus" on the solution of the global problem. In this setting the performance of the chosen protocol naturally depends on the network topology, since to solve the problem each agent *has to* communicate and receive information from all other agents. In particular, the number of iterations required by decentralised iterative gradient methods typically scales with the inverse of the spectral gap of the communication matrix (a.k.a. gossip or consensus matrix) [18, 42, 43], which reflects the performance of gossip protocols in the problem of distributed averaging [9, 17, 44, 4].

Many distributed machine learning problems, in particular those involving empirical risk minimisation, have been framed in the context of consensus optimisation. However, as highlighted in [46] and more recently in [38], often these problems have more structure than consensus optimisation due to the statistical regularity of the data. When the agents' functions are the empirical risk of their local data, in the setting where the local data comes from the *same* unknown distribution (homogeneous setting), the functions held by each agent are similar to one another by the phenomenon of statistical concentration. In particular, in the limit of an infinite amount of data per agent, the local functions are the same and agents do *not* need to communicate to solve the problem. This phenomenon highlights the existence of a natural trade-off between statistics and communication. While statistical similarities of local objective functions and the statistics/communication trade-off have been investigated and exploited in centralised server-clients setup, typically in the analysis and design of divide-and-conquer schemes [60, 28, 20, 32, 26, 1, 62, 46, 45, 61, 2], only recently there has been some investigation into the interplay between statistics and communication/network-topology in the decentralised setting. The authors in [6] investigate the interplay between the spectral norm of the data-generating distribution and the inverse spectral gap of the communication matrix for Distributed Stochastic Gradient Descent in the case of strongly convex losses. As most of the literature on decentralised machine learning, this work also focuses on minimising the training error and not the test/prediction error (numerical experiments are given for the test error). Some works have investigated the performance on the test loss in the single-pass/online stochastic setting where agents use each data point only once. The authors in [37, 51] investigate a distributed regularised online learning setting [55] and obtain guarantees for a "multi-step" Distributed Stochastic Mirror Descent algorithm where agents reach consensus on their stochastic gradients in-between computation steps. The works [25] and [3] consider the performance of Distributed Stochastic Gradient Descent algorithms in the non-convex smooth case. They investigate the average performance of the agents over the network in terms of convergence to a stationary point of the test loss [19] and show that a linear speed-up in computational time can be achieved provided the number of samples seen, equivalently the number of iterations performed, exceeds the network size times the inverse of the spectral gap, each raised to a certain power. The work [38] seems to be the first to have considered minimisation of the test error in the multi-pass/offline stochastic setting that more naturally relates to the classical literature on consensus optimisation. The authors investigate stability of Distributed Stochastic Gradient Descent on the test error and show that for smooth and convex losses the number of iterations required to achieve optimal statistical rates scales with the inverse of the spectral gap of the gossip matrix, a term that captures the noise of the gradients' estimates, and a term that controls the statistical proximity of the local empirical losses.

1.1 Contributions

In this work we investigate the implicit-regularisation learning performance of full-batch Distributed Gradient Descent [33] on the test error in the context of non-parametric regression with the square loss function. In the homogeneous setting where agents hold independent and identically distributed data points, we investigate the choice of step size and number of iterations that guarantee each agent to achieve optimal statistical rates with respect to all the samples in the network. We build a theoretical framework that allows to directly and explicitly exploit the statistical concentration of quantities (i.e. batched gradients) held by agents. On the one hand, exploiting concentration yields savings on computation, i.e. it allows to achieve faster convergence rates compared to methods that do not exploit

concentration in their parameter tuning. On the other hand, it yields savings on communication, as it allows to take advantage of the trade-off between statistical power and communication costs. Firstly, we show that if agents hold sufficiently many samples with respect to the network size, then Distributed Gradient Descent achieves optimal statistical rates up to poly-logarithmic factors with a number of iterations that scales with the inverse of the spectral gap of the communication matrix divided by the number of samples owned by each agent raised to a problem-dependent power, up to a statistics-induced threshold. Previous results for decentralised iterative gradient schemes in the context of consensus optimisation do not take advantage of the statistical nature of decentralised empirical risk minimisation problems. In the statistical setting that we consider, these methods would require a larger number of iterations that scales only with respect to the inverse of the spectral gap. Secondly, we show that if agents additionally hold sufficiently many samples with respect to the inverse of the spectral gap, then the *same* order of iterations allows Distributed Gradient Descent and Single-Machine Gradient Descent (i.e. gradient descent run on a single machine that holds all the samples in the network) to achieve optimal statistical rates up to poly-logarithmic factors. Provided the communication delay is sufficiently small, this yields a *linear* speed-up in runtime over Single-Machine Gradient Descent, with a "single-step" method that performs a single communication round per local gradient descent step. Single-step methods that do not exploit concentration can only achieve a linear speed-up in runtime in graphs with spectral gap bounded away from zero, i.e. expanders or the complete graph. Our results demonstrate how the increased statistical similarity between the local empirical risk functions can make up for a decreased connectivity in the graph topology, showing that a linear speed-up in runtime can be achieved in *any* graph topology by exploiting concentration. To the best of our knowledge, we seem to be the first to isolate this type of phenomena.

We prove our results under the standard "source" and "capacity" assumptions in non-parametric regression. These assumptions relate, respectively, to the projection of the optimal predictor on the hypothesis space and to the effective dimension of this space [59, 12]. A contribution of this work is to show that proper tuning yields, up to poly-logarithmic terms, optimal non-parametric rates in decentralised learning. As far as we aware, in the distributed setting such guarantees have been established only for centralised divide-and-conquer methods [60, 28, 20, 32, 26].

To prove our results we build upon previous work for Single-Machine Gradient Descent applied to non-parametric regression, in particular the line of works [57, 40, 27]. Exploiting that in our setting the iterates of Distributed Gradient Descent can be written in terms of products of linear operators depending on the data held by agents, we decompose the excess risk into bias and sample variance terms for Single-Machine Gradient Descent plus an additional quantity that captures the error incurred by using a decentralised protocol over the communication network. We analyse this network error term by further decomposing it into a term that behaves similarly to the consensus error previously considered in [18, 33], and a new higher-order term. We control both terms by using the structure of the gradient updates, which allows us to analyse the interplay between statistics, via concentration, and network topology, via mixing of random walks related to the gossip matrix.

The work is structured as follows. Section 2 presents the setting, assumptions, and algorithm that we consider. Section 3 states the main convergence result and discusses implications from the point of view of statistics, computation and communication. Section 4 presents the error decomposition into bias, variance, and network error, and it illustrates the implicit regularisation strategy that we adopt. Section 5 highlights some of the features of our contribution in the light of future research directions. The appendix in the supplementary material is structured as follows. Section A includes some remarks about our results. Section B illustrates the main scheme of the proofs, highlighting the interplay between statistics and network topology. Section C contains the full details of the proofs.

2 Setup

In this section we describe the learning problem, assumptions and algorithm that we consider.

2.1 Learning problem: decentralised non-parametric least-squares regression

We adopt the setting used in [40, 27], which involves regression in abstract Hilbert spaces. This setting is of relevance for applications related to the Reproducing Kernel Hilbert Space (RKHS). See the work in [57] and references therein.

Let H be a separable Hilbert Space with inner product and induced norm denoted by $\langle \cdot, \cdot \rangle_H$ and $\| \cdot \|_H$, respectively. Let $X \subseteq H$ be the input space and $Y \subset \mathbb{R}$ be the output space. Let ρ be an unknown probability measure on $Z = X \times Y$, $\rho_X(\cdot)$ be the marginal on X, and $\rho(\cdot|x)$ be the conditional distribution on Y given $x \in X$. Assume that there exists a constant $\kappa \in [1, \infty)$ so that

$$\langle x, x' \rangle_H \leq \kappa^2, \quad \forall x, x' \in X. \tag{1}$$

Let the network of agents be modelled by a simple, connected, undirected, finite graph $G = (V, E)$, with $|V| = n$ nodes joined by edges $E \subseteq V \times V$. Edges represent communication constraints: agents $v, w \in V$ can only communicate if they share an edge $(v, w) \in E$. We consider the homogeneous setting where each agent $v \in V$ is given m data points $\mathbf{z}_v := \{\mathbf{x}_v, \mathbf{y}_v\}$ sampled independently from ρ, where $\mathbf{x}_v = \{x_{i,v}\}_{i=1,\dots,m}$ and $\mathbf{y}_v = \{y_{i,v}\}_{i=1,\dots,m}$, and each pair $(x_{i,v}, y_{i,v})$ is sampled from ρ. The problem under study is the minimisation of the test/prediction risk with the square loss:

$$\inf_{\omega \in H} \mathcal{E}(\omega), \qquad \mathcal{E}(\omega) = \int_{X \times Y} (\langle \omega, x \rangle_H - y)^2 d\rho(x, y), \tag{2}$$

The quality of an approximate solution $\widehat{\omega} \in H$ is measured by the excess risk $\mathcal{E}(\widehat{\omega}) - \inf_{\omega \in H} \mathcal{E}(\omega)$.

Notation Given a matrix $A \in \mathbb{R}^{n \times n}$, let A_{vw} denote the (v, w)-th element and $A_v = (A_{vw})_{w=1,\dots,n}$ denote the v-th row. Let $O(\cdot)$ denote orders of magnitudes up to constants in n and m, and $\widetilde{O}(\cdot)$ denote orders of magnitudes up to both constants and poly-logarithmic terms in n and m. Let $\lesssim, \gtrsim, \simeq$ denote inequalities and equalities modulo constants and poly-logarithmic terms in n, m. We use the notation $a \vee b = \max\{a, b\}$ and $a \wedge b = \min\{a, b\}$.

2.2 Assumptions

The assumptions that we consider are standard in non-parametric regression [27, 35]. The first assumption is a control on the even moments of the response.

Assumption 1. *There exist $M \in (0, \infty)$ and $\nu \in (1, \infty)$ such that for any $\ell \in \mathbb{N}$ we have $\int_Y y^{2\ell} d\rho(y|x) \leq \nu \ell! M^\ell$ ρ_X-almost surely.*

Let $L^2(H, \rho_X)$ be the Hilbert space of square-integrable functions from H to $\mathbb{R}$ with respect to ρ_X, with norm $\|f\|_\rho := (\int_X |f(x)|^2 d\rho_X(x))^{1/2}$. Let $\mathcal{L}_\rho : L^2(H, \rho_X) \to L^2(H, \rho_X)$ be the operator defined as $\mathcal{L}_\rho(f) := \int_X \langle x, \cdot \rangle_H f(x) d\rho_X(x)$. Under Assumption 1 the operator $\mathcal{L}_\rho$ can be proved to be in the class of positive trace operators [15], and therefore the r-th power $\mathcal{L}_\rho^r$, with $r \in \mathbb{R}$, can be defined by using spectral theory. Let us also define the operator $\mathcal{T}_\rho : H \to H$ as $\mathcal{T}_\rho := \int_X \langle x, \cdot \rangle_H x d\rho_X(x)$ and its operator norm $\|\mathcal{T}_\rho\| := \sup_{\omega \in H, \|\omega\|_H = 1} \|\mathcal{T}_\rho \omega\|_H$. The function minimising the expected squared loss (2) over all measurable functions $f : H \to \mathbb{R}$ is known to be the conditional expectation $f_\rho(x) := \int_Y y d\rho(y|x)$ for $x \in \mathcal{X}$. Let $H_\rho := \{f : X \to \mathbb{R} \,|\, \exists \omega \in H \text{ with } f(x) = \langle w, x \rangle_H, \rho_X\text{-almost surely}\}$ be the hypothesis space that we consider. The optimal f_ρ may not be in H_ρ as under Assumption 1 the space of functions searched H_ρ is a subspace of $L^2(H, \rho_X)$. Let f_H denote the projection of f_ρ onto the closure of H_ρ in $L^2(H, \rho_X)$. Searching for a solution to (2) is equivalent to searching for a linear function in H_ρ that approximates f_H.
The following assumption quantifies how well the target function f_H can be approximated in H_ρ.

Assumption 2. *There exist $r > 0$ and $R > 0$ such that $\|\mathcal{L}_\rho^{-r} f_H\|_\rho \leq R$.*

This assumption is often called the "source" condition [12]. Representing f_H in the eigenspace of $\mathcal{L}_\rho$, this condition can be related to the rate at which the coefficients of this representation decay. The bigger r is, the faster the decay, and more stringent the assumption is. In particular, if $r \geq 1/2$ then the target function is in the hypothesis space $f_H \in H_\rho$. The last assumption is on the capacity of the hypothesis space.

Assumption 3. *There exist $\gamma \in (0, 1], c_\gamma > 0$ such that $\mathrm{Tr}(\mathcal{L}_\rho(\mathcal{L}_\rho + \lambda I)^{-1}) \leq c_\gamma \lambda^{-\gamma}$ for all $\lambda > 0$.*

Assumption 3 relates to the effective dimension of the underlying regression problem [59, 12] and is often called the "capacity" assumption. This assumption is always satisfied for $\gamma = 1$ and $c_\gamma = \kappa^2$ since $\mathcal{L}_\rho$ is a trace class operator. This case is called the capacity-independent setting. Meanwhile, this assumption is satisfied for $\gamma \in (0, 1]$ if, for instance, the eigenvalues of $\mathcal{L}_\rho$, denoted by $\{\tau_i\}_{i \geq 1}$, decay sufficiently quickly, i.e. $\tau_i = O(i^{-1/\gamma})$. This case allows improved rates to be obtained. For more details on the interpretation of these assumptions we refer to the work in [40, 27, 35].

2.3 Algorithm: distributed gradient descent

We now describe the Distributed Gradient Descent algorithm [33] and its application to the problem of non-parametric regression. Let $P \in \mathbb{R}^{n\times n}_{\geq 0}$ be a symmetric doubly-stochastic matrix, i.e. $P = P^\top$ and $P\mathbf{1} = \mathbf{1}$ where $\mathbf{1} = (1, \dots, 1) \in \mathbb{R}^n$ is the vector of all ones. Let P be supported on the graph, i.e. for any $v \neq w$, $P_{vw} \neq 0$ only if $(v, w) \in E$. The matrix P encodes local averaging on the network: when each agent has a real number represented by the vector $a = (a_v)_{v\in V} \in \mathbb{R}^n$, the vector $(Pa)_v = \sum_{w\in V} P_{vw}a_w$ for $v \in V$ encodes what each agent computes after taking a weighted average of its own and neighbours' numbers. Distributed Gradient Descent is implemented by communication on the network through the gossip matrix P. Initialised at $w_{1,v} = 0$ for $v \in V$, the iterates of the Distributed Gradient Descent are defined as follows, for $v \in V$ and $t \geq 1$:

$$\omega_{t+1,v} = \sum_{w\in V} P_{vw}\Big(\omega_{t,w} - \eta_t \frac{1}{m}\sum_{i=1}^{m}\big(\langle \omega_{t,w}, x_{i,w}\rangle_H - y_{i,w}\big)x_{i,w}\Big), \tag{3}$$

where $\{\eta_t\}_{t\geq 1}$ is the sequence of positive step sizes. The iterates (3) can be seen as a combination of two steps: first, each agent $w \in V$ performs a local gradient descent step $\omega_{t+1/2,w} = \omega_{t,w} - \eta_t \frac{1}{m}\sum_{i=1}^{m}\big(\langle \omega_{t,w}, x_{i,w}\rangle_H - y_{i,w}\big)x_{i,w}$; second, each agent performs local averaging through the consensus step[1] $\omega_{t+1,v} = \sum_{w\in V} P_{vw}\omega_{t+1/2,w}$. We treat gradient descent as a statistical device. We are interested in tuning the parameters of the algorithm to bound the expected value of the excess risk $\mathbf{E}[\mathcal{E}(\omega_{t+1,v})] - \inf_{\omega\in H}\mathcal{E}(\omega)$, where $\mathbf{E}[\,\cdot\,]$ denotes expectation with respect to the data $\{\mathbf{z}_v\}_{v\in V}$.

Network dependence Let σ_2 be the second largest eigenvalue in magnitude of the communication matrix P. Specifically, given the spectral decomposition of the gossip matrix $P = \sum_{l=1}^{n}\lambda_l u_l u_l^\top$ where $1 = \lambda_1 \geq \lambda_2 \geq, \dots, \geq \lambda_n > -1$ are the ordered real eigenvalues of P and $\{u_l\}_{l=1,\dots,n}$ the associated eigenvectors, we have $\sigma_2 := \max\{|\lambda_2|, |\lambda_n|\}$. In many settings, the spectral gap scales with the size of the network raised to a certain power depending on the topology. For instance, supposing G is a finite regular graph and the communication matrix is the random walk matrix, then the inverse of the spectral gap $(1-\sigma_2)^{-1}$ scales as $\Theta(1)$ for a complete graph, $\Theta(n)$ for a grid, and $\Theta(n^2)$ for a cycle [14, 24, 18]. The question of designing gossip matrices P that yield better (smaller) scaling for the quantity $(1-\sigma_2)^{-1}$ has been investigated [56], and it has been found numerically that the rates mentioned above can not be improved unless lifted graphs are considered [44].

3 Main result: optimal statistical rates with linear speed-up in runtime

We now state and highlight the main contribution of this work in the context of decentralised statistical optimisation. The result that we are about to state in Theorem 1 showcases the interplay between statistics and communication that arise from the statistical regularities of the problem. This result shows the existence of a "big data" regime where Distributed Gradient Descent can achieve a linear (in the number of agents n) speed-up in runtime compared to Single-Machine Gradient Descent.

Theorem 1. *Let Assumptions 1, 2, 3 hold with $r \geq 1/2$ and $2r + \gamma > 2$. Let t be the smallest integer greater than the quantity*

$$\underbrace{(nm)^{1/(2r+\gamma)}}_{\textit{Single-Machine Iterations}} \times \begin{cases} \left(\frac{(nm)^{2r/(2r+\gamma)}}{m(1-\sigma_2)^\gamma}\right)^{1/\gamma} \vee 1 & \textit{if } m \geq n^{2r/\gamma} \\ \frac{(nm)^{r/(2r+\gamma)}}{\sqrt{m}(1-\sigma_2)} & \textit{otherwise} \end{cases}$$

Let $\eta_s \equiv \eta = \frac{\kappa^{-2}(nm)^{1/(2r+\gamma)}}{t}$ $\forall s \geq 1$. If $m \geq n^{\frac{2r+2+\gamma}{2r+\gamma-2}}$ and $n \geq 2(1+r)\log(\frac{n}{1-\sigma_2})$, then $\forall v \in V$:

$$\mathbf{E}[\mathcal{E}(\omega_{t+1,v})] - \inf_{\omega\in H}\mathcal{E}(\omega) \leq C(nm)^{-2r/(2r+\gamma)},$$

where C depends on $\kappa^2, \|\mathcal{T}_\rho\|, M, \nu, r, R, \gamma, c_\gamma$, and polynomials of $\log(nm)$ and $\log(\frac{1}{1-\sigma_2})$.

[1] We note, while this assumes agents communicate infinite dimensional quantities in the general non-parametric setting, the framework we consider accommodates finite approximations of infinite dimensional quantities whilst accounting for the statistical precision [13].

Theorem 1 shows that when agents are given sufficiently many samples (m) with respect to the number of agents (n), $m \geq n^{\frac{2r+2+\gamma}{2r+\gamma-2}}$, proper tuning of the step size and number of iterations (a form of implicit regularisation) allows Distributed Gradient Descent to recover the optimal statistical rate $O((nm)^{-2r/(2r+\gamma)})$ for $r \in (1/2, 1)$ [12] up to poly-logarithmic terms.

Single-Machine Gradient Descent run on all of the observations has been previously shown to reach optimal statistical accuracy with a number of iterations of the order $t_{\text{Single-Machine}} \sim O((nm)^{1/(2r+\gamma)})$ [27]. The number of iterations $t \equiv t_{\text{Distributed}}$ prescribed by Theorem 1 scales like $t_{\text{Single-Machine}}$ times a network-dependent factor that is a function of the inverse of the spectral gap $(1-\sigma_2)^{-1}$. The fact that the number of iterations required to reach a prescribed level of error accuracy is inversely proportional to the spectral gap is a standard feature of iterative gradient methods applied to generic decentralised consensus optimisation problems [18, 42, 43]. This dependence encodes the fact that in the case of *generic* objective functions assigned to agents, agents *have to* share information with everyone to solve the global problem and minimise the sum of the local functions; hence, more iterations are required in graph topologies that are less well-connected. In the present homogeneous setting, however, the statistical nature of the problem allows to exploit concentration of random variables to characterise the existence of a (network-dependent) "big data" regime where the number of iterations does *not* depend on the network topology. The trade-off between statistics and communication is encoded by the dependence of the tuning parameters (stopping time and step size) on the number of samples m assigned to each agent. Observe that the factor $(\frac{(nm)^{2r/(2r+\gamma)}}{m(1-\sigma_2)^{\gamma}})^{1/\gamma} \vee 1$ is a decreasing function of m, up to the threshold 1. When $m \geq \frac{n^{2r/\gamma}}{(1-\sigma_2)^{2r+\gamma}} \vee n^{\frac{2r+2+\gamma}{2r+\gamma-2}}$ this factor becomes 1 and Theorem 1 guarantees that the *same* order of iterations allows both Distributed and Single-Machine Gradient Descent to achieve the optimal statistical rates up to poly-logarithmic factors. This regime represents the case when the increased statistical similarity between the local empirical risk functions assigned to each agent (increasing as a function of m, as described by the non-asymptotic Law of Large Numbers) makes up for the decreased connectivity in the graph topology (typically decreasing with the spectral gap $1-\sigma_2$) to yield a linear speed-up in runtime over Single-Machine Gradient Descent when the communication delay between agents is sufficiently small. See Section 3.1 below.

The result of Theorem 1 depends on some other requirements which we now briefly discuss. The requirement $n \geq 2(1+r)\log(\frac{n}{1-\sigma_2})$ is technical and arises from the need to perform sufficiently many iterations to reach the mixing time of the gossip matrix P, i.e. $t \gtrsim (1-\sigma_2)^{-1}$. Noting that the number of iterations t depends on the number of agents, samples and spectral gap. The requirement $2r+\gamma > 2$ relates to the difficulty of the estimation problem and is stronger than a similar condition seen for single-machine gradient methods where $2r+\gamma > 1$, see for instance the works [27, 35]. This requirement, alongside $m \geq n^{\frac{2r+2+\gamma}{2r+\gamma-2}}$, ensures that the higher-order error terms arising from considering a decentralised protocol decay sufficiently quickly with respect to the number of samples owned by agents m. The condition $m \geq n^{\frac{2r+2+\gamma}{2r+\gamma-2}}$ can be removed if the covariance operator $\mathcal{T}_\rho$ is assumed to be known to agents, which aligns with the additive noise oracle in single-pass Stochastic Gradient Descent [16] or fixed-design regression in finite-dimensional settings [21]. The condition $m \geq n^{2r/\gamma}$ corresponds to the case when the rate of concentration of the batched gradients held by agents (i.e. $1/m$) is faster than the optimal statistical rate, i.e. $\frac{1}{m} \leq (nm)^{-2r/(2r+\gamma)}$. This condition becomes more stringent (i.e. more data per agent is needed) as the problem becomes easier from a statistical point of view and r and $1/\gamma$ increase (see discussion in Section 2.2). This is due to the fact that as r and $1/\gamma$ increase, only the statistical rate improves while the rate of concentration in the network error stays the same, implying that more data is needed to balance the two terms.

3.1 Linear speed-up in runtime

Let gradient computations cost 1 unit of time and communication delay between agents be τ units of time.[2] Denote the number of iterations required by Single-Machine Gradient Descent and Distributed Gradient Descent to achieve the optimal statistical rate by $t_{\text{Single-Machine}}$ and $t_{\text{Distributed}}$, respectively. The speed-up in computational time obtained by running the distributed protocol over the single-machine protocol is of the order $\frac{t_{\text{Single-Machine}}}{t_{\text{Distributed}}} \frac{nm}{m+\tau+\text{Deg}(P)}$, where $\text{Deg}(P) = \max_{v \in V} |\{P_{vw} \neq 0, w \in V\}|$ is the maximum degree of the communication matrix P. Theorem 1 implies that when $m \geq$

[2] For details on this communication model as well as comparison to [50] see remarks within Appendix A.

$\frac{n^{2r/\gamma}}{(1-\sigma_2)^{2r+\gamma}} \vee n^{\frac{2r+2+\gamma}{2r+\gamma-2}}$ then $t_{\text{Distributed}} \sim t_{\text{Single-Machine}}$, and if $\tau + \text{Deg}(P)$ grows as $O(m)$ then the speed-up in computational time is of order n, linear in the number of agents. Classical "single-step" decentralised methods that alternate single communication rounds per local gradient computation, such as the methods inspired by [33], do not exploit concentration and have a runtime that scales with the inverse of the spectral gap, without any threshold. As a result, these methods only yield a linear speed-up in graphs with spectral gap bounded away from zero, i.e. expanders or the complete graph. See below for more details. On the other hand, "multi-step" methods that alternate multiple communication rounds per local gradient computation, such as the ones considered in [37, 51, 42, 43], display a runtime that scales with a factor of the form $m + \frac{\tau+\text{Deg}(P)}{1-\sigma_2}$ in our setting. Thus, while these methods can achieve a linear speed-up in any graph topology in the "big data" regime $m \gtrsim \frac{\tau+\text{Deg}(P)}{1-\sigma_2}$ without exploiting concentration, they require an additional amount of communication rounds that is network-dependent and scales with the inverse of the spectral gap. For a cycle graph, for instance, this means an extra $O(n^2)$ communication steps per iteration (or $O(n)$ for gossip-accelerated methods). Hence, classical decentralised optimisation methods that do not exploit concentration suffer from a trade-off between runtime and communication cost: if you reduce the first you increase the second, and viceversa. Our results show that single-step methods can achieve a linear speed-up in runtime in *any* graph topology by exploiting concentration: statistics allows to find a regime where it is possible to simultaneously have a linear speed-up in runtime without increasing communication.

Comparison to single-step decentralised methods that do not exploit concentration Decentralised optimisation methods that do not consider statistical concentration rates in their parameter tuning can not exploit the statistics/communication trade-off encoded by the presence of the factor $(\frac{(nm)^{2r/(2r+\gamma)}}{m(1-\sigma_2)^{\gamma}})^{1/\gamma} \vee 1$ in Theorem 1, and they typically require a smaller step size and more iterations to achieve optimal statistical rates. The convergence rate typically achieved by classical consensus optimisation methods, e.g. [18], is recovered in Theorem 1 when $m = n^{2r/\gamma}$ as in this case the number of iterations required becomes $t \sim \frac{(nm)^{1/(2r+\gamma)}}{1-\sigma_2}$, which corresponds to $t_{\text{Single-Machine}}$ scaled by a certain power of $1/(1-\sigma_2)$ (in our setting the power is 1). This represents the setting where the choice of step size aligns with the choice in the single-machine case scaled by $(1-\sigma_2)$, and a linear speed-up occurs when $(1-\sigma_2)^{-1} = O(1)$. Since the network error is decreasing in m in our case (due to concentration), larger step sizes can be chosen for $m > n^{2r/\gamma}$. Specifically, the single-machine step size is now scaled by $[(1-\sigma_2)(\frac{m}{n^{2r/\gamma}})^{1/(2r+\gamma)}] \vee 1$, yielding a linear speed-up when $(1-\sigma_2)^{-1} = O((\frac{m}{n^{2r/\gamma}})^{1/(2r+\gamma)})$, which, as m increases, is a weaker requirement on the network topology over the standard consensus optimisation setting.

4 General result: error decomposition and implicit regularisation

Theorem 1 is a corollary of the next result, which explicitly highlights the interplay between statistics and network topology and the implicit regularisation role of the step size and number of iterations.

Theorem 2. *Let Assumptions 1, 2, 3 hold with $r \geq 1/2$. Let $\eta_s = \eta s^{-\theta}$ $\forall s \geq 1$ with $\theta \in (0, 3/4)$ and $\eta \in (0, \kappa^{-2}]$. If $t/2 \geq \lceil \frac{(r+1)\log(t)}{1-\sigma_2} \rceil =: t^\star$, then for all $v \in V$, $\alpha \in [0, 1/2]$ and $\gamma' \in [1, \gamma]$:*

$$\mathbf{E}[\mathcal{E}(\omega_{t+1,v})] - \inf_{\omega \in H} \mathcal{E}(\omega)$$

$$\leq \left[q_1 (\eta t^{1-\theta})^{-2r} + q_2 (nm)^{-2r/(2r+\gamma)} \left(1 \vee (nm)^{-2/(2r+\gamma)} (\eta t^{1-\theta})^2 \vee t^{-2} (\eta t^{1-\theta})^2 \right) \right] \log^2(t) \quad (4)$$

$$+ q_3 \frac{\log^2(n) \log^2(t^\star)}{m} \left(\eta^2 t^{-2r} \vee (m^{-1} (\eta t^\star)^{1+2\alpha}) \vee (\eta t^\star)^{\gamma'+2\alpha} \right) \quad (5)$$

$$+ q_4 \frac{\log^4(n) \log^2(t)}{m^2} \left(1 \vee (\eta t^{1-\theta})^2 \vee t^{-2} (\eta t^{1-\theta})^4 \right) \left((m^{-1} \eta t^{1-\theta}) \vee (\eta t^{1-\theta})^{\gamma} \right) \quad (6)$$

where q_1, q_2, q_3, q_4 are all constants depending on $\kappa^2, \|\mathcal{T}_\rho\|, M, \nu, r, R, \gamma, c_\gamma$.

The bound in Theorem 2 shows that the excess risk has been decomposed into three main terms, as detailed in Section B.1. The first term (4) corresponds to the error achieved by Single-Machine Gradient Descent run on all nm samples. It consists of both bias and sample variance terms [27]. The second two terms (5) and (6) characterise the network error due to the use of a decentralised

protocol. These terms decrease with the number of samples m owned by each agent. This captures the fact that, as agents are given samples from the *same* unknown distribution, agents are in fact solving the same learning problem and their local empirical loss functions concentrate to the same objective as m increases. The decentralised error term is itself composed of two terms which decay at different rates with respect to m. The term in (5) is dominant and decays at the order of $\widetilde{O}(1/m)$. This can be interpreted as the consensus error seen in the works [33, 18] for instance. As in that setting, this quantity is also increasing with the step size η and decreasing with the spectral gap of the communication matrix $1-\sigma_2$, as encoded by $t^\star$. The term (6) decays at the faster rate of $\widetilde{O}(1/m^2)$. This is a higher-order error term that is not appearing in the error decomposition when the covariance operator $\mathcal{T}_\rho$ is assumed to be known to agents. This quantity arises from the interaction between the local averaging on the network through P and what has been previously labelled as the "multiplicative" noise in the single-machine single-pass stochastic gradient setting for least squares [16], i.e. the empirical covariance operator interacting with the iterates at each step. Section B.2 provides a high-level illustration of the analysis of the Network Error terms (5) and (6).

The bound in Theorem 2 shows how the algorithmic parameters—step size and number of iterations—act as regularisation parameters for Distributed Gradient Descent, following what is seen in the single-machine setting. Theorem 1 demonstrates how optimal statistical rates can be recovered by tuning these parameters appropriately with respect to the network topology, network size, number of samples, and with respect to the estimation problem itself. The bound in Theorem 1 is obtained from the bound in Theorem 2 by first tuning the quantity ηt to the order $(nm)^{1/(2r+\gamma)}$ so that the bias and variance terms in (4) achieve the optimal statistical rate. This leaves the tuning of the remaining degree of freedom (say η) to ensure that also the network error achieves the optimal statistical rate. The high-level idea is the following. As m increases, the network error is dominated by the term in (5) that is proportional to the factor $(\eta t^\star)^{\gamma'+2\alpha}/m$. There are two ways to choose the largest possible step size η to guarantee that this factor is $\widetilde{O}((nm)^{-2r/(2r+\gamma)})$, depending on whether the rate of concentration of the batched gradients held by agents is faster than the optimal statistical rate or not, i.e., whether $m \geq n^{2r/\gamma}$ is true or not (cf. Section 3). The two cases yield the factors $(\frac{(nm)^{2r/(2r+\gamma)}}{m(1-\sigma_2)^\gamma})^{1/\gamma} \vee 1$ and $\frac{(nm)^{r/(2r+\gamma)}}{\sqrt{m}(1-\sigma_2)}$ in Theorem 1, corresponding to the choice $\gamma' = \gamma\,,\ \alpha = 0$ and $\gamma' = 1\,,\ \alpha = 1/2$, respectively. If the concentration of the batched gradients held by agents fully compensates for the network error, i.e. $m \geq \frac{n^{2r/\gamma}}{(1-\sigma_2)^{2r+\gamma}}$, then $(\eta t^\star)^{\gamma'+2\alpha}/m \simeq (nm)^{-2r/(2r+\gamma)}$ with a constant step size and $t_{\text{Distributed}} \sim t_{\text{Single-Machine}} \sim (nm)^{1/(2r+\gamma)}$, yielding the regime where a linear speed-up occurs. For more details on the parameters α, γ', see Lemma 8 in Appendix C.3.1.

5 Future directions

We highlight some of the features of our contribution and outline directions for future research.

Non-parametric setting We prove bounds in the attainable case $r \geq 1/2$. The non-attainable case $r < 1/2$ is known to be more challenging [27], and it is natural to investigate to what extent our results can be extended to that setting. We consider the case $\gamma > 0$ which does not include the finite-dimensional setting $H = \mathbb{R}^d$, $\gamma = 0$, where the optimal rate is $O(d/(nm))$ [54]. While adapting our results to this setting requires minor modifications, optimal bounds would only hold for "easy" estimation problems with $r > 1$ due to the higher-order term in the network error. Improvements require getting better bounds on this term, potentially using a different learning rate.

General loss functions The analysis that we develop is specific to the square loss, which yields the bias/variance error decomposition and allows to get explicit characterisations by expanding the squares. While the concentration phenomena that we exploit are generic, different techniques are required to extend our analysis to other losses, as in the single-machine setting. The statistical proximity of agents' functions in the finite-dimensional setting has been investigated in [38].

Statistics/communication trade-off with sparse/randomised gossip In this work we show that when agents hold sufficiently many samples, then Distributed and Single-Machine Gradient Descent achieve the optimal statistical rate with the same order of iterations. This motivates balancing and trading off communication and statistics, e.g., investigating statistically robust procedures in settings when agents communicate with a subset of neighbours, either deterministically or randomly [9, 17, 4].

Stochastic gradient descent and mini-batches Our work exploits concentration of gradients around their means, so full-batch gradients (i.e. batches of size m) yield the concentration rate $1/m$. In single-machine learning, stochastic gradient descent [39] has been shown to achieve good statistical performance in a variety of settings while allowing for computational savings. Extending our findings to stochastic methods with appropriate mini-batch sizes is another venue for future investigation.

Acknowledgments

Dominic Richards is supported by the EPSRC and MRC through the OxWaSP CDT programme (EP/L016710/1). Patrick Rebeschini is supported in part by the Alan Turing Institute under the EPSRC grant EP/N510129/1. We would like to thank Francis Bach, Lorenzo Rosasco and Alessandro Rudi for helpful discussions.

References

[1] Alekh Agarwal and John C. Duchi. Distributed delayed stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 873–881, 2011.

[2] Yossi Arjevani and Ohad Shamir. Communication complexity of distributed convex learning and optimization. In *Advances in neural information processing systems*, pages 1756–1764, 2015.

[3] Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Michael Rabbat. Stochastic gradient push for distributed deep learning. *arXiv preprint arXiv:1811.10792*, 2018.

[4] Florence Bénézit, Alexandros G Dimakis, Patrick Thiran, and Martin Vetterli. Order-optimal consensus through randomized path averaging. *IEEE Transactions on Information Theory*, 56(10):5150–5167, 2010.

[5] Raphaël Berthier, Francis Bach, and Pierre Gaillard. Accelerated Gossip in Networks of Given Dimension using Jacobi Polynomial Iterations. *arXiv preprint arXiv:1805.08531*, May 2018.

[6] Avleen S Bijral, Anand D Sarwate, and Nathan Srebro. Data-dependent convergence for consensus stochastic optimization. *IEEE Transactions on Automatic Control*, 62(9):4483–4498, 2017.

[7] Gilles Blanchard and Nicole Mücke. Optimal rates for regularization of statistical inverse learning problems. *Foundations of Computational Mathematics*, 18(4):971–1013, 2018.

[8] Olivier Bousquet and Léon Bottou. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems*, pages 161–168, 2008.

[9] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE transactions on information theory*, 52(6):2508–2530, 2006.

[10] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011.

[11] Ming Cao, Daniel A Spielman, and Edmund M Yeh. Accelerated gossip algorithms for distributed computation. In *Proc. of the 44th Annual Allerton Conference on Communication, Control, and Computation*, pages 952–959. Citeseer, 2006.

[12] Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

[13] Luigi Carratino, Alessandro Rudi, and Lorenzo Rosasco. Learning with sgd and random features. In *Advances in Neural Information Processing Systems*, pages 10192–10203, 2018.

[14] Fan R.K. Chung and Fan Chung Graham. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

[15] Felipe Cucker and Ding Xuan Zhou. *Learning theory: an approximation theory viewpoint*, volume 24. Cambridge University Press, 2007.

[16] Aymeric Dieuleveut, Nicolas Flammarion, and Francis Bach. Harder, better, faster, stronger convergence rates for least-squares regression. *The Journal of Machine Learning Research*, 18(1):3520–3570, 2017.

[17] Alexandros DG Dimakis, Anand D Sarwate, and Martin J Wainwright. Geographic gossip: Efficient averaging for sensor networks. *IEEE Transactions on Signal Processing*, 56(3):1205–1216, 2008.

[18] John C. Duchi, Alekh Agarwal, and Martin J. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic Control*, 57(3):592–606, 2012.

[19] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[20] Zheng-Chu Guo, Shao-Bo Lin, and Ding-Xuan Zhou. Learning theory of distributed spectral algorithms. *Inverse Problems*, 33(7):074009, 2017.

[21] László Györfi, Michael Kohler, Adam Krzyzak, and Harro Walk. *A distribution-free theory of nonparametric regression*. Springer Science & Business Media, 2006.

[22] Bjorn Johansson, Maben Rabi, and Mikael Johansson. A simple peer-to-peer algorithm for distributed optimization in sensor networks. In *Decision and Control, 2007 46th IEEE Conference on*, pages 4705–4710. IEEE, 2007.

[23] Björn Johansson, Maben Rabi, and Mikael Johansson. A randomized incremental subgradient method for distributed optimization in networked systems. *SIAM Journal on Optimization*, 20(3):1157–1170, 2009.

[24] David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

[25] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 5330–5340, 2017.

[26] Junhong Lin and Volkan Cevher. Optimal convergence for distributed learning with stochastic gradient methods and spectral-regularization algorithms. *arXiv preprint arXiv:1801.07226*, 2018.

[27] Junhong Lin and Lorenzo Rosasco. Optimal rates for multi-pass stochastic gradient methods. *Journal of Machine Learning Research*, 18(97):1–47, 2017.

[28] Shao-Bo Lin, Xin Guo, and Ding-Xuan Zhou. Distributed learning with regularized least squares. *The Journal of Machine Learning Research*, 18(1):3202–3232, 2017.

[29] Ilan Lobel and Asuman Ozdaglar. Distributed subgradient methods for convex optimization over random networks. *IEEE Transactions on Automatic Control*, 56(6):1291–1306, 2011.

[30] Ion Matei and John S Baras. Performance evaluation of the consensus-based distributed subgradient method under random communication topologies. *IEEE Journal of Selected Topics in Signal Processing*, 5(4):754–771, 2011.

[31] Aryan Mokhtari and Alejandro Ribeiro. Dsa: Decentralized double stochastic averaging gradient algorithm. *Journal of Machine Learning Research*, 17(61):1–35, 2016.

[32] Nicole Mücke and Gilles Blanchard. Parallelizing spectrally regularized kernel algorithms. *The Journal of Machine Learning Research*, 19(1):1069–1097, 2018.

[33] Angelia Nedić, Alex Olshevsky, Asuman Ozdaglar, and John N. Tsitsiklis. On distributed averaging algorithms and quantization effects. *IEEE Transactions on Automatic Control*, 54(11):2506–2517, 2009.

[34] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[35] Loucas Pillaud-Vivien, Alessandro Rudi, and Francis Bach. Statistical optimality of stochastic gradient descent on hard learning problems through multiple passes. In *Advances in Neural Information Processing Systems 31*, pages 8125–8135. 2018.

[36] IF Pinelis and AI Sakhanenko. Remarks on inequalities for large deviation probabilities. *Theory of Probability & Its Applications*, 30(1):143–148, 1986.

[37] M. Rabbat. Multi-agent mirror descent for decentralized stochastic optimization. In *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 517–520, Dec 2015.

[38] D. Richards and P. Rebeschini. Graph-Dependent Implicit Regularisation for Distributed Stochastic Subgradient Descent. *ArXiv e-prints*, sep 2018.

[39] Herbert Robbins and Sutton Monro. A stochastic approximation method. In *Herbert Robbins Selected Papers*, pages 102–109. Springer, 1985.

[40] Lorenzo Rosasco and Silvia Villa. Learning with incremental iterative regularization. In *Advances in Neural Information Processing Systems*, pages 1630–1638, 2015.

[41] Ali H. Sayed. Adaptive networks. *Proceedings of the IEEE*, 102(4):460–497, 2014.

[42] Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3027–3036. JMLR. org, 2017.

[43] Kevin Scaman, Francis Bach, Sébastien Bubeck, Laurent Massoulié, and Yin Tat Lee. Optimal algorithms for non-smooth distributed optimization in networks. In *Advances in Neural Information Processing Systems*, pages 2745–2754, 2018.

[44] Devavrat Shah. Gossip algorithms. *Foundations and Trends® in Networking*, 3(1):1–125, 2009.

[45] Ohad Shamir. Fundamental limits of online and distributed algorithms for statistical learning and estimation. In *Advances in Neural Information Processing Systems*, pages 163–171, 2014.

[46] Ohad Shamir and Nathan Srebro. Distributed stochastic optimization and learning. In *Communication, Control, and Computing (Allerton), 2014 52nd Annual Allerton Conference on*, pages 850–857. IEEE, 2014.

[47] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

[48] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[49] Pierre Tarres and Yuan Yao. Online learning as stochastic approximation of regularization paths: Optimality and almost-sure convergence. *IEEE Trans. Information Theory*, 60(9):5716–5735, 2014.

[50] Konstantinos Tsianos, Sean Lawlor, and Michael G Rabbat. Communication/computation tradeoffs in consensus-based distributed optimization. In *Advances in neural information processing systems*, pages 1943–1951, 2012.

[51] Konstantinos I Tsianos and Michael G Rabbat. Efficient distributed online prediction and stochastic optimization with approximate distributed averaging. *IEEE Transactions on Signal and Information Processing over Networks*, 2(4):489–506, 2016.

[52] John Tsitsiklis, Dimitri Bertsekas, and Michael Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE transactions on automatic control*, 31(9):803–812, 1986.

[53] John Nikolas Tsitsiklis. Problems in decentralized decision making and computation. Technical report, Massachusetts Inst Of Tech Cambridge Lab For Information And Decision Systems, 1984.

[54] Alexandre B Tsybakov. Optimal rates of aggregation. In *Learning Theory and Kernel Machines*, pages 303–313. Springer, 2003.

[55] Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(Oct):2543–2596, 2010.

[56] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems and Control Letters*, 53(1):65–78, 2004.

[57] Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

[58] Yiming Ying and Massimiliano Pontil. Online gradient descent learning algorithms. *Foundations of Computational Mathematics*, 8(5):561–596, 2008.

[59] Tong Zhang. Learning bounds for kernel regression using effective data dimensionality. *Neural Computation*, 17(9):2077–2098, 2005.

[60] Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *The Journal of Machine Learning Research*, 16(1):3299–3340, 2015.

[61] Yuchen Zhang and Xiao Lin. Disco: Distributed optimization for self-concordant empirical loss. In *International conference on machine learning*, pages 362–370, 2015.

[62] Yuchen Zhang, Martin J. Wainwright, and John C. Duchi. Communication-efficient algorithms for statistical optimization. In *Advances in Neural Information Processing Systems*, pages 1502–1510, 2012.

A Remarks

In this section we present some remarks about our work.

Alternative protocol The protocol investigated in [33] updates the iterates via $\omega_{t+1,v} = \sum_{w\in V} P_{vw}\omega_{t,w} - \eta_t \frac{1}{m}\sum_{i=1}^{m}\big(\langle \omega_{t,v}, x_{i,v}\rangle_H - y_{i,v}\big)x_{i,v}$. The original motivations for this protocol are that it is fully decentralised, that agents are only required communicate locally, and that it reduces to a distributed averaging consensus protocol when the gradient is zero. The protocol (3) that we consider preserves these properties while making the analysis easier. For a discussion on the difference between the two protocols we refer to [41].

Network error The network error terms (5) and (6) track the error between the distributed protocol and the ideal single-machine protocol. In the case of a complete graph the deviation is zero so the network terms vanish and the convergence rates for Single-Machine Gradient Descent are recovered. Following the literature on decentralised optimisation, we present our final results (cf. Theorem 2) in terms of the spectral gap, so plugging in the spectral gap of a complete graph in the bound in Theorem 2 does not immediately yield the Single-Machine Gradient Descent result.

Parameter tuning The choice of parameters in Theorem 1 depends on the quantities r and γ that are related to the estimation problem. In practice, these quantities are often unknown. In the single-machine setting, this lack of knowledge is typically addressed via cross-validation [48]. Investigating the design of decentralised cross-validation schemes is outside of the scope of this work and we leave it to future research. However, we highlight that as we consider implicit regularisation strategies and, in particular, early stopping, model complexity can be controlled with iteration time and this yields computational savings for cross-validation compared to methods that required to solve independent problem instances for different choices of parameters.

Accelerated gossip Accelerated gossip schemes can also be considered to yield improved dependence on the network topology, depending on the amount of information agents have access to about the communication matrix P. Accelerated gossip can be achieved by replacing the matrix P by a polynomial of appropriate order, e.g. k, leading to $\widetilde{P} := \sum_{\ell=1}^{k}\alpha_\ell P^\ell$. The weights $\{\alpha\}_{\ell=1,\dots,K}$ can be tuned to increase the spectral gap i.e. $(1-\sigma_2(\widetilde{P}))^{-1} \leq (1-\sigma_2)^{-1}$. We highlight that the algorithm that we consider only needs to have access to the number of nodes n and the second largest eigenvalue in magnitude σ_2 of the matrix P. Within this framework, one can use Chebyshev polynomials to obtain the improved rate $(1-\sigma_2(\widetilde{P}))^{-1/2}$, and more information on the spectrum of P yields better rates on the transitive phase [11, 5].

Additional requirements in Theorem 2 Theorem 2 includes two additional requirements over single-machine gradient descent, which we briefly explain the origins of. The requirement $\theta \leq 3/4$ is purely cosmetic and serves to yield a cleaner bound. For more details, see the proof of Lemma 9 in Section C.3.2. The requirement $t/2 \geq \frac{(r+1)\log(t)}{1-\sigma_2}$, on the other hand, often arises when analysing Distributed Gradient Descent, see [18] for instance. In particular, it ensures sufficient iterations have been performed to reach the mixing time of the Markov chain associated to P. See Section C.3.1.

Communication model We include additional details on the communication model. Consider a lockstep communication model where each round lasts for τ units of time. Within each round, agents send/receive the messages to/from their neighbours in order to implement a single update of algorithm (3). With a gradient evaluation costing 1 unit of time, each iteration of Distributed Gradient Descent takes the following amount of time $m + \tau + \mathrm{Deg}(P)$: m gradient evaluations; τ in communication delay; $\mathrm{Deg}(P)$ for each agent to aggregating their neighbours and own gradients, as the sum in algorithm (3) $\sum_{w\in V} P_{vw}$ has computational cost $O(\mathrm{Deg}(P))$. The delay τ can depend on factors arising from: noisy transmission, compressing or decompressing messages and synchronizing with neighbours. One particular model for τ is studied within [50] and discussed in the following remark.

Comparison to speed-up and communication model within [50] The work [50] assumes the delay τ is a linear function of the network degree and some transmit time $T_{\mathrm{Transmit}} \geq 0$ so $\tau = T_{\mathrm{Transmit}}\mathrm{Deg}(P)$. In our work, for sufficiently many samples m, the speed-up under this model for

any network topology is of the order $\frac{nm}{m+\text{Deg}(P)T_{\text{Transmit}}}$. Meanwhile, the speed-up seen within [50] is[3] of the order $\frac{nm}{m+\text{Deg}(P)T_{\text{Transmit}}}(1-\sigma_2)$, that is, same as ours but scaled by the spectral gap of the communication matrix P.

B Proof scheme

In this section we illustrate the main scheme for the proof of Theorem 2, from which Theorem 1 follows. Section B.1 presents the error decomposition into bias, variance, and network terms. Section B.2 presents the sketch of the statistical analysis for these terms, which is given in full in Section C.

B.1 Error decomposition

The error decomposition is based on the introduction of two auxiliary processes used to compare the iterates of Distributed Gradient Descent (3).

The first auxiliary process represents the iterates generated if agents were to know the marginal distribution ρ_X. Initialised at $\mu_1 = 0$, the process is defined as follows for $t \geq 1$:

$$\mu_{t+1} = \mu_t - \eta_t \int_X (\langle \mu_t, x\rangle_H - f_\rho(x)) x d\rho_X(x).$$

This device has already been used in the analysis of non-parametric regression in the single-machine setting [27].

The second auxiliary process represents the iterates generated if agents were to be part of a complete graph topology and were to use the protocol given by $P = \frac{1}{n}\mathbf{1}\mathbf{1}^\top$. Initialised at $\xi_{1,v} = 0$ for all $v \in V$, the process is defined as follows for $t \geq 1$:

$$\xi_{t+1,v} = \sum_{w\in V} \frac{1}{n}\bigg(\xi_{t,w} - \eta_t \frac{1}{m}\sum_{i=1}^m (\langle \xi_{t,w}, x_{i,w}\rangle_H - y_{i,w})x_{i,w}\bigg).$$

The analysis of iterative decentralised algorithms typically builds upon the introduction of a device analogous to this one [33, 18]. Initialised at $\xi_1 = 0$, Single-Machine Gradient Descent is defined as follows for $t \geq 1$:

$$\xi_{t+1} = \xi_t - \eta_t \frac{1}{nm}\sum_{w\in V}\sum_{i=1}^m \big(\langle \xi_t, x_{i,w}\rangle_H - y_{i,w}\big)x_{i,w}.$$

It is easy to see that we have $\xi_{t,v} = \xi_t$ for $t \geq 1$ and $v \in V$. This allows us to produce an analysis of Distributed Gradient Descent that relies upon known results for Single-Machine Gradient Descent.

Let us introduce the linear map $\mathcal{S}_\rho : H \to L^2(H, \rho_X)$ defined by $\mathcal{S}_\rho \omega = \langle \omega, \cdot\rangle_H$. The following error decomposition holds.

Proposition 1. *For any $t \geq 1$ and $v \in V$ we have*

$$\mathcal{E}(\omega_{t,v}) - \inf_{\omega\in H}\mathcal{E}(\omega) \leq 2\underbrace{\|\mathcal{S}_\rho\mu_t - f_H\|_\rho^2}_{(\textbf{\textit{Bias}})^2} + 4\underbrace{\|\mathcal{S}_\rho(\xi_t - \mu_t)\|_\rho^2}_{\textbf{\textit{Sample Variance}}} + 4\underbrace{\|\mathcal{S}_\rho(\omega_{t,v} - \xi_{t,v})\|_\rho^2}_{\textbf{\textit{Network Error}}}.$$

Proof. From the work in [40], $\mathcal{E}(\omega) - \inf_{\omega\in H}\mathcal{E}(\omega) = \|\mathcal{S}_\rho\omega - f_H\|_\rho^2$ for any $\omega \in H$. Adding and subtracting $\mathcal{S}_\rho\mu_t$ and using $\|x-y\|_\rho^2 \leq (\|x\|_\rho + \|y\|_\rho)^2 \leq 2\|x\|_\rho^2 + 2\|y\|_\rho^2$ we get

$$\mathcal{E}(\omega_{t,v}) - \inf_{\omega\in H}\mathcal{E}(\omega) = \|\mathcal{S}_\rho\omega_{t,v} - \mathcal{S}_\rho\mu_t + \mathcal{S}_\rho\mu_t - f_H\|_\rho^2 \leq 2\|\mathcal{S}_\rho\omega_{t,v} - \mathcal{S}_\rho\mu_t\|_\rho^2 + 2\|\mathcal{S}_\rho\mu_t - f_H\|_\rho^2.$$

Following the same steps, adding and subtracting $\mathcal{S}_\rho\xi_{t,v}$, we find

$$\|\mathcal{S}_\rho\omega_{t,v} - \mathcal{S}_\rho\mu_t\|_\rho^2 = \|\mathcal{S}_\rho\omega_{t,v} - \mathcal{S}_\rho\xi_{t,v} + \mathcal{S}_\rho\xi_{t,v} - \mathcal{S}_\rho\mu_t\|_\rho^2 \leq 2\|\mathcal{S}_\rho(\omega_{t,v} - \xi_{t,v})\|_\rho^2 + 2\|\mathcal{S}_\rho(\xi_t - \mu_t)\|_\rho^2$$

where we used the equality of $\{\xi_{s,v}\}_{s\geq 1}$ and $\{\xi_s\}_{s\geq 1}$. □

[3] The units of time within [50, Section 3.2] are in terms of the time taken to compute a gradient for nm samples, and as such, can be translated into units per gradient computation by multiplying by nm.

Proposition 1 decomposes the error into three terms. The first term $\|\mathcal{S}_\rho \mu_t - f_H\|_\rho^2$ is deterministic and corresponds to the square of the **Bias** in the single-machine setting [57]. The second term $\|\mathcal{S}_\rho(\xi_t - \mu_t)\|_\rho^2$ aligns with what is called the **Sample Variance** in the single-machine setting, and in this case matches the sample variance obtained for Single-Machine Gradient Descent run on all nm observations. The third term $\|\mathcal{S}_\rho(\omega_{t,v} - \xi_{t,v})\|_\rho^2$ accounts for the error due to performing a decentralised protocol and we call it the **Network Error**.

B.2 Statistical analysis of error terms

In this section we illustrate the main ideas of the statistical analysis used to control the error terms in Proposition 1. Full details are given in Section C.

Notation Let t and k be positive natural numbers with $t-1 \geq k \geq 1$. For any operator $\mathcal{L} : H \to H$, define $\Pi_{t:k+1}(\mathcal{L}) := (I - \eta_t \mathcal{L})(I - \eta_{t-1}\mathcal{L}) \cdots (I - \eta_{k+1}\mathcal{L})$, with the convention $\Pi_{t:t+1}(\mathcal{L}) := I$, where I is the identity operator on H. Let $w_{t:k+1} \equiv w_t w_{t-1} \dots w_{k+1} := (w_t, w_{t-1}, \dots, w_{k+1}) \in V^{t-k}$ denote a sequence of nodes in V. For a family of operators indexed by the nodes on the graph $\{\mathcal{L}_v\}_{v \in V}$, define $\mathcal{L}_{w_{t:k+1}} := (\mathcal{L}_{w_t}, \dots, \mathcal{L}_{w_{k+1}})$ and $\Pi_{t:k+1}(\mathcal{L}_{w_{t:k+1}}) := (I - \eta_t \mathcal{L}_{w_t})(I - \eta_{t-1}\mathcal{L}_{w_{t-1}}) \cdots (I - \eta_{k+1}\mathcal{L}_{w_{k+1}})$, with $\Pi_{t:t+1}(\mathcal{L}_{w_{t:t+1}}) := I$. Let $P_{w_{t:k+1}} := P_{w_t w_{t-1}} P_{w_{t-1} w_{t-2}} \cdots P_{w_{k+2} w_{k+1}}$ be the probability of the path generated by a Markov Chain with transition kernel P. For each agent $v \in V$, let $\mathcal{T}_{\mathbf{x}_v} : H \to H$ with $\mathcal{T}_{\mathbf{x}_v} = \frac{1}{m}\sum_{i=1}^m \langle \cdot, x_{i,v} \rangle_H x_{i,v}$ be the empirical covariance operator associated to the agent's own data $\mathbf{x}_v$, and let $\mathcal{T}_{\mathbf{x}_{w_{t:k+1}}} := (\mathcal{T}_{\mathbf{x}_{w_t}}, \dots, \mathcal{T}_{\mathbf{x}_{w_{k+1}}})$. For $k \geq 1, v \in V$, let $N_{k,v} \in H$ be a random variable that only depends on the randomness in $\mathbf{z}_v$ and that has zero mean, $\mathbf{E}[N_{k,v}] = 0$. The random variable $N_{k,v}$, formally defined in (8) in Section C.3, captures the sampling error introduced at iteration k of gradient descent by agent v. For the discussion below it suffices to mentioned the two above properties.

The following paragraphs discuss the analysis for each of the error terms.

Bias The analysis follows the single-machine setting and is given in Proposition 2 in Section C.1.

Sample Variance The analysis follows the single-machine setting [27], although the original result yields a high probability bound with a requirement on the number of samples nm. We therefore follow the result in [26] which yields a bound in high probability without a condition on the sample size. The bound for this term is presented in Theorem 3 in Section C.2.

Network Error Unraveling the iterates (Lemma 5 in Section C.3) we get, for any $v \in V, t \geq 1$:

$$\|\mathcal{S}_\rho(\omega_{t+1,v} - \xi_{t+1,v})\|_\rho = \left\| \sum_{k=1}^t \eta_k \sum_{w_{t:k} \in V^{t-k+1}} \left(P_{v w_{t:k}} - \frac{1}{n^{t-k+1}} \right) \mathcal{T}_\rho^{1/2} \Pi_{t:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t:k+1}}}) N_{k,w_k} \right\|_H .$$

This characterisation makes explicit the dependence of the network error on both the communication protocol used by the agents, via the dependence on the mixing properties of the gossip matrix P along each path $v w_{t:k}$, and on the statistical properties of the problem, via the product of empirical covariance operators held by the agents along each path $w_{t:k+1}$. As the randomness in the quantities N_{k,w_k} might depend on the randomness in the empirical covariance operators, we further decompose the network error into two terms so that we can use the property $\mathbf{E}[N_{k,w_k}] = 0$. By adding and subtracting the terms $\Pi_{t:k+1}(\mathcal{T}_\rho)$ inside the sums we have

$$\|\mathcal{S}_\rho(\omega_{t+1,v} - \xi_{t+1,v})\|_\rho^2 \leq 2 \underbrace{\left\| \sum_{k=1}^t \eta_k \sum_{w_{t:k} \in V^{t-k+1}} \left(P_{v w_{t:k}} - \frac{1}{n^{t-k+1}} \right) \mathcal{T}_\rho^{1/2} \Pi_{t:k+1}(\mathcal{T}_\rho) N_{k,w_k} \right\|_H^2}_{(\textbf{Population Covariance Error})^2}$$

$$+ 2 \underbrace{\left\| \sum_{k=1}^t \eta_k \sum_{w_{t:k} \in V^{t-k+1}} \left(P_{v w_{t:k}} - \frac{1}{n^{t-k+1}} \right) \mathcal{T}_\rho^{1/2} \left(\Pi_{t:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t:k+1}}}) - \Pi_{t:k+1}(\mathcal{T}_\rho) \right) N_{k,w_k} \right\|_H^2}_{(\textbf{Residual Empirical Covariance Error})^2} .$$

From a statistical point of view, the **Population Covariance Error** term only depends on the population covariance via the quantities $\Pi_{t:k+1}(\mathcal{T}_\rho)$, and the only source of randomness is given by N_{k,w_k}. Using concentration for N_{k,w_k}, the square of this error term can be bounded by a quantity that decreases as $\widetilde{O}(1/m)$, as announced in Section 4 alongside the discussion of Theorem 2. On the other hand, the **Residual Empirical Covariance Error** term depends on *deviations* between the empirical covariance and the population covariance via the quantities $\Pi_{t:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t:k+1}}}) - \Pi_{t:k+1}(\mathcal{T}_\rho)$. Exploiting the additional concentration of these factors allows us to bound the square of this error term by a higher-order quantity that decreases as $\widetilde{O}(1/m^2)$.

We now present a separate discussion on the analysis for these two error terms, emphasizing the interplay between network topology (mixing of random walks on graphs) and statistics (concentration). The final bound for the network error is presented in Theorem 4 in Section C.3.

Population Covariance Error Expanding the square yields a summation over all pairs of paths:

$$\Big\| \sum_{k=1}^{t} \sum_{w_{t:k} \in V^{t-k+1}} a_{k,w_{t:k}} \Big\|_H^2 = \sum_{k,k'=1}^{t} \sum_{w_{t:k} \in V^{t-k+1}} \sum_{w'_{t:k'} \in V^{t-k'+1}} \langle a_{k,w_{t:k}} a_{k',w'_{t:k'}} \rangle_H$$

for properly defined quantities $a_{k,w_{t:k}}$ (the dependence on v is neglected). When taking the expectation, as the random variables $\{N_{k,v}\}_{k\geq 1, v\in V}$ have zero mean and are independent across agents $v \in V$, the only paths left are those that intersect at the final node, i.e. $w_{t:k}, w'_{t:k'}$ such that $w_k = w_{k'}$. Moreover, as all agents have identically distributed data, the remaining expectation no longer depends on the final node of the paths. The remaining quantity is then analysed by bounding the probability of the two paths intersecting at the final node in terms of the second largest eigenvalue in magnitude of P and by bounding the inner product by the norm product. This yields

$$\mathbf{E}[(\textbf{Pop. Cov. Error})^2] \leq \mathbf{E}\bigg[\bigg(\sum_{k=1}^{t} \sigma_2^{t-k+1} \eta_k \|\mathcal{T}_\rho^{1/2} \Pi_{t:k+1}(\mathcal{T}_\rho) N_{k,v}\|_H \bigg)^2 \bigg].$$

Denoting the mixing time associated to P as $t^\star$, the series is divided into well-mixed and poorly-mixed terms, respectively, $k \leq t - t^\star$ and $k \geq t - t^\star$. The well-mixed terms are controlled by σ_2^{t-k+1}. Meanwhile, for the poorly-mixed terms begin by taking for $\lambda > 0$ $\max_{k=1,\dots,t} \big\{ \|(\mathcal{T}_\rho + \lambda I)^{-1/2} N_{k,v}\|_H^2 \big\}$ outside of the series. The expectation of this maximum is controlled through concentration and becomes $\widetilde{O}(\frac{1}{m^2\lambda} + \frac{1}{m\lambda^{\gamma'}})$ for $\gamma' \in [1, \gamma]$. The remaining series is controlled through the contraction of the term $\|\mathcal{T}_\rho^{1/2} \Pi_{t:k+1}(\mathcal{T}_\rho)(\mathcal{T}_\rho + \lambda I)^{1/2}\|$ and choosing $\lambda \simeq 1/(\eta t^\star)$. These two steps lead to this term being of the order $O(\frac{\eta t^\star}{m^2} + \frac{(\eta t^\star)^{\gamma'}}{m})$, which dominates the well-mixed terms and contributes to the dependence on the inverse of the spectral gap of P. The free parameter $\gamma' \in [1, \gamma]$ is left open as a smaller step size η is used to control this term when $m \leq n^{2r/\gamma}$. The final bound is given in Lemma 8 in Section C.3.1.

Residual Empirical Covariance Error The analysis of this term is based on the following identity (Proposition 5 in Section C.3.2), for any $t - 1 \geq k$ and any $w_{t:k+1} \in V^{t-k}$:

$$\Pi_{t:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t:k+1}}}) - \Pi_{t:k+1}(\mathcal{T}_\rho) = \sum_{j=k+1}^{t} \eta_j \Pi_{t:j+1}(\mathcal{T}_\rho)(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_{w_j}}) \Pi_{j-1:k+1}(\mathcal{T}_{\mathbf{x}_{w_{j-1:k+1}}}).$$

The above decomposition has two key properties. Firstly, it depends upon differences between the empirical covariance operators $\mathcal{T}_{\mathbf{x}_{w_j}}$ and its expectation $\mathcal{T}_\rho$. This allows concentration to be used, and, alongside the concentration for $N_{k,v}$, it ensures that (**Resid. Emp. Cov. Error**)^2 is of order $\widetilde{O}(1/m^2)$. Secondly, it is of the form $\sum_{j=k+1}^{t} \eta_j \Pi_{t:j+1}(\mathcal{T}_\rho)[\cdots]$, where $[\cdots]$ indicates the right most factors and the quantity shown aligns with the filter function for gradient descent [26, Example 2]. Once again the contractive property of the quantity $\Pi_{t:j+1}(\mathcal{T}_\rho)$ allows to give sharper rates with respect to the step size and number of iterations. Without it, the choice of step size $\eta_t = \eta t^{-\theta}$ would yield a bound for (**Resid. Emp. Cov. Error**)^2 of the order $\big(\sum_{k=1}^{t} \eta_k \sum_{j=k+1}^{t-1} \eta_j\big)^2 \simeq (\eta t^{1-\theta})^4$. The contraction allows to show that (**Resid. Emp. Cov. Error**)^2 grows at the reduced order $(\eta t^{1-\theta})^3$,

and the addition of the capacity assumption allows it to be further reduced to the order $(\eta t^{1-\theta})^{2+\gamma}$. The final high-probability bound is given in Lemma 9 in Section C.3.2. This being stronger than the bound in expectation required for Theorem 2.

C Proofs

Before going on to present proofs for the main result some notation is introduced following [40, 27]. Some notation is repeated from the previous sections, as additional details are included. Adopt the convention for sums $\sum_{k=t+1}^{t} = 0$. For a given bounded operator $\mathcal{L} : L^2(H, \rho_X) \to H$, let $\|\mathcal{L}\|$ denote the operator norm of $\mathcal{L}$, i.e. $\|\mathcal{L}\| = \sup_{f \in L^2(H,\rho_X), \|f\|_\rho = 1} \|\mathcal{L}f\|_H$. Let $\mathcal{S}_\rho : H \to L^2(H, \rho_X)$ be the linear map $\omega \to \langle \omega, \cdot \rangle_H$,which is bounded by κ under Assumption 1. Consider the adjoint operator $\mathcal{S}_\rho^\star : L^2(H, \rho_X) \to H$, the covariance operator $\mathcal{T}_\rho : H \to H$ given by $\mathcal{T}_\rho = \mathcal{S}_\rho^\star \mathcal{S}_\rho$, and the operator $\mathcal{L}_\rho : L^2(H, \rho_X) \to L^2(H, \rho_X)$ given by $\mathcal{L}_\rho = \mathcal{S}_\rho \mathcal{S}_\rho^\star$. We have $\mathcal{S}_\rho^\star g = \int_X x g(x) d\rho_X(x)$ and $\mathcal{T}_\rho = \int_X \langle \cdot, x \rangle_H x d\rho_X(x)$. For any $\omega \in H$ the following isometry property holds [48]

$$\|\mathcal{S}_\rho \omega\|_\rho = \|\sqrt{\mathcal{T}_\rho}\omega\|_H.$$

The following notation was utilised in the analysis of Single-Machine Gradient Descent [40, 27]. In this case it aligns with all of the observations in the network $\mathbf{y} := \{y_{i,v}\}_{i=1,\dots,m, v \in V} \in \mathbb{R}^{m|V|}$ and $\mathbf{x} = \{x_{i,v}\}_{i=1,\dots,m, v \in V}$. Define the sampling operator $\mathcal{S}_\mathbf{x} : H \to \mathbb{R}^{m|V|}$ by $\big(\mathcal{S}_\mathbf{x}\omega\big)_{(i,v)} = \langle \omega, x_{i,v} \rangle_H$, for $i = 1, \dots, m, v \in V$. Let $\| \cdot \|_{\mathbb{R}^{m|V|}}$ denote the Euclidean norm in in $\mathbb{R}^{m|V|}$ times the factor $1/\sqrt{nm}$. Its adjoint operator $\mathcal{S}_\mathbf{x}^\star : \mathbb{R}^{m|V|} \to H$, defined by $\langle \mathcal{S}_\mathbf{x}^\star \mathbf{y}, \omega \rangle_H = \langle \mathbf{y}, \mathcal{S}_\mathbf{x}\omega \rangle_{\mathbb{R}^{m|V|}}$ for $\mathbf{y} \in \mathbb{R}^{m|V|}$, is given by $\mathcal{S}_\mathbf{x}^\star \mathbf{y} = \frac{1}{nm} \sum_{v \in V} \sum_{i=1}^m y_{i,v} x_{i,v}$. Define the covariance operator with respect to all of the samples $\mathcal{T}_\mathbf{x} : H \to H$ such that $\mathcal{T}_\mathbf{x} = \mathcal{S}_\mathbf{x}^\star \mathcal{S}_\mathbf{x}$. We have

$$\mathcal{T}_\mathbf{x} = \frac{1}{nm} \sum_{v \in V} \sum_{i=1}^m \langle \cdot, x_{i,v} \rangle_H x_{i,v}.$$

The following notation is analogous to the single-machine notation just introduced, although now with respect to the datasets held by individual agents, i.e. $\mathbf{x}_v$ and $\mathbf{y}_v$ for $v \in V$. Let $\mathcal{S}_{\mathbf{x}_v} : H \to \mathbb{R}^m$ with $(\mathcal{S}_{\mathbf{x}_v}\omega)_i = \langle \omega, x_{i,v} \rangle_H$ for $i = 1, \dots, m$. Let $\| \cdot \|_{\mathbb{R}^m}$ be the Euclidean norm in $\| \cdot \|_{\mathbb{R}^m}$ times $1/\sqrt{m}$. Its adjoint operator $\mathcal{S}_{\mathbf{x}_v}^\star : \mathbb{R}^m \to H$, defined by $\langle \mathcal{S}_{\mathbf{x}_v}^\star \mathbf{y}_v, \omega \rangle_H = \langle \mathbf{y}_v, \mathcal{S}_{\mathbf{x}_v}\omega \rangle_{\mathbb{R}^m}$ for $\mathbf{y}_v \in \mathbb{R}^m$, is given by $\mathcal{S}_{\mathbf{x}_v}^\star \mathbf{y}_v = \frac{1}{m} \sum_{i=1}^m y_{i,v} x_{i,v}$. The empirical covariance operator $\mathcal{T}_{\mathbf{x}_v} : H \to H$ is such that $\mathcal{T}_{\mathbf{x}_v} = \mathcal{S}_{\mathbf{x}_v}^\star \mathcal{S}_{\mathbf{x}_v}$, with $\mathcal{T}_{\mathbf{x}_v} = \frac{1}{m} \sum_{i=1}^m \langle \cdot, x_{i,v} \rangle_H x_{i,v}$.

Using this notation, the processes $\{\mu_t\}_{t \geq 1}$, $\{\omega_{t,v}\}_{t \geq 1}$, and $\{\xi_t\}_{t \geq 1}$ can be rewritten as follows. The population process reads

$$\mu_{t+1} = \mu_t - \eta_t \big(\mathcal{T}_\rho \mu_t - \mathcal{S}_\rho^\star f_\rho\big).$$

The gossiped process reads

$$\omega_{t+1,v} = \sum_{w \in V} P_{vw} \Big(\omega_{t,w} - \eta_t \big(\mathcal{T}_{\mathbf{x}_w} \omega_{t,w} - \mathcal{S}_{\mathbf{x}_w}^\star \mathbf{y}_w\big)\Big).$$

The single-machine process reads

$$\xi_{t+1} = \xi_t - \eta_t \big(\mathcal{T}_\mathbf{x} \xi_t - \mathcal{S}_\mathbf{x}^\star \mathbf{y}\big).$$

The next three sections present bounds for the three error terms introduced in Proposition 1. Section C.1 presents a bound for the Bias term, which follows directly from the results in [27] and references therein. Section C.2 establishes a bound for the Sample Variance term, which follows from results in [26]. Section C.3 develops bounds for the Network Error term, which are a novel contribution of this work. Section C.4 brings the results of the previous three sections together to establish the proofs of Theorem 2 and Theorem 1, respectively. Section C.5 includes useful inequalities that are needed to establish our results.

C.1 Bias

The following bound on the Bias term $\|\mathcal{S}_\rho \mu_t - f_H\|_\rho^2$ is taken from [27], inspired by [57, 40].

Proposition 2. *[27, Appendix C Proposition 2] Under Assumption 2, let $\eta\kappa^2 \leq 1$. Then for any $t \in \mathbb{N}$,*

$$\|\mathcal{S}_\rho \mu_t - f_H\|_\rho \leq R\bigg(\frac{r}{2\sum_{j=1}^t \eta_j}\bigg)^r.$$

In particular, if $\eta_t = \eta t^{-\theta}$ for all $t \in \mathbb{N}$, with $\eta \in (0, \kappa^{-2}]$ and $\theta \in [0,1)$ then

$$\|\mathcal{S}_\rho \mu_t - f_H\|_\rho \leq R r^r \eta^{-r} t^{r(\theta-1)}.$$

C.2 Sample Variance

In this section we establish a bound for the expectation of the Sample Variance term $\mathbf{E}[\|\mathcal{S}_\rho(\xi_t - \mu_t)\|_\rho^2]$. The following lemma summaries a number of intermediary steps in [27] for bounding the Sample Variance term. It arises from representing the iterates $\{\xi_t - \mu_t\}_{t\geq 1}$ in terms of the stochastic sequence $\{N_k\}_{k\geq 1}$ which characterises the sample noise introduced in the iterations of gradient descent. These terms are controlled via the empirical covariance operator $\mathcal{T}_\mathbf{x}$ and the population covariance operator $\mathcal{T}_\rho$ while introducing the pseudo-regularisation parameter $\lambda > 0$ and utilising the contractive property of the gradient updates. For the following, let us introduce the notation $\mathcal{T}_{\rho,\lambda} = \mathcal{T}_\rho + \lambda I$ and $\mathcal{T}_{\mathbf{x},\lambda} = \mathcal{T}_\mathbf{x} + \lambda I$.

Lemma 1. *Let $\eta_1 \kappa^2 \leq 1$ and $0 \leq \lambda$. For any $t \in \mathbb{N}$ we have*

$$\begin{aligned}
&\|\mathcal{S}_\rho(\xi_{t+1} - \mu_{t+1})\|_\rho \\
&\leq \bigg(\sum_{k=1}^{t-1} \frac{\eta_k \|\mathcal{T}_{\rho,\lambda}^{-1/2} N_k\|_H}{2\sum_{i=k+1}^t \eta_i} + \lambda \sum_{k=1}^{t-1} \eta_k \|\mathcal{T}_{\rho,\lambda}^{-1/2} N_k\|_H + \|\mathcal{T}_\rho\|^{1/2}(\|\mathcal{T}_\rho\| + \lambda)^{1/2} \eta_t \|\mathcal{T}_{\rho,\lambda}^{-1/2} N_t\|_H\bigg) \\
&\quad \times \|\mathcal{T}_{\mathbf{x},\lambda}^{-1/2} \mathcal{T}_\rho^{1/2}\| \|\mathcal{T}_{\mathbf{x},\lambda}^{-1/2} \mathcal{T}_{\rho,\lambda}^{1/2}\|,
\end{aligned}$$

where

$$N_k = (\mathcal{T}_\rho \mu_k - \mathcal{S}_\rho^\star f_\rho) - (\mathcal{T}_\mathbf{x} \mu_k - \mathcal{S}_\mathbf{x}^\star \mathbf{y}), \quad \forall k \in \mathbb{N}. \tag{7}$$

Proof. The proof of this result follows the proof of [27, Proposition 3]. □

The two quantities left to control are $\|\mathcal{T}_{\rho,\lambda}^{-1/2} N_k\|_H$ for $k \in \mathbb{N}$ as well as $\|(\mathcal{T}_\mathbf{x} + \lambda I)^{-1/2} \mathcal{T}_\rho^{1/2}\|^2$. The first of these quantities is controlled by [27, Lemma 18] which is summarised in the following lemma.

Lemma 2. *[27, Lemma 18] Let Assumptions 1, 2, 3 hold with $r \geq 1/2$ and $\{N_k\}_{k\geq 1}$ be as in (7). For any $\lambda > 0$, with probability at least $1-\delta$, the following holds $\forall k \in \mathbb{N}$*

$$\|(\mathcal{T}_\rho + \lambda I)^{-1/2} N_k\|_H \leq 4(R\kappa^{2r} + \sqrt{M})\bigg(\frac{\kappa}{nm\sqrt{\lambda}} + \frac{\sqrt{2\sqrt{\nu} c_\gamma}}{\sqrt{nm\lambda^\gamma}}\bigg) \log\frac{4}{\delta}.$$

The next lemma from [26, Lemma 19 Remark 1] controls $\|(\mathcal{T}_\mathbf{x} + \lambda I)^{-1/2} \mathcal{T}_\rho^{1/2}\|^2$.

Lemma 3. *[26, Lemma 19, Remark 1] Let $\delta \in (0,1)$ and $\lambda = (nm)^{-p}$ for some $p \geq 0$. With probability at least $1-\delta$ the following holds*

$$\begin{aligned}
&\|\mathcal{T}_\rho^{1/2}(\mathcal{T}_\mathbf{x} + \lambda)^{-1/2}\|^2 \leq \|(\mathcal{T}_\rho + \lambda I)^{1/2}(\mathcal{T}_\mathbf{x} + \lambda)^{-1/2}\|^2 \\
&\leq 24\kappa^2\bigg(\log\frac{4\kappa^2(c_\gamma+1)}{\delta\|\mathcal{T}_\rho\|} + p\gamma \min\bigg(\frac{1}{e(1-p)_+}, \log nm\bigg)\bigg)(1 \vee (nm)^{p-1}).
\end{aligned}$$

Bringing together the three previous results yields the following high-probability bound for the Sample Variance term.

Proposition 3. *Fix $\delta \in (0,1)$ and $p \in (0,1)$. Let Assumptions 1, 2 and 3 hold with $r \geq 1/2$ and $\eta_t = \eta t^{-\theta}$ with $\eta\kappa^2 \leq 1$, $\theta \in [0,1)$. The following holds with probability at least $1-\delta$ for any $t \in \mathbb{N}$*

$$\begin{aligned}
&\|\mathcal{S}_\rho(\xi_{t+1} - \mu_{t+1})\|_\rho \\
&\leq \widetilde{d}_1 \min\Big(\frac{1}{e(1-p)_+}, \log nm\Big)\frac{\log(t)}{(nm)^{(1-p\gamma)/2}}(1 \vee (nm)^{-p}\eta t^{1-\theta} \vee \eta t^{-\theta}) \log^2\frac{\widetilde{d}_2}{\delta},
\end{aligned}$$

with $\widetilde{d}_1 = 768\frac{\kappa^2\|\mathcal{T}_\rho\|^{1/2}(\|\mathcal{T}_\rho\|+1)^{1/2}(R\kappa^{2r}+\sqrt{M})(\kappa+\sqrt{2\sqrt{\nu}c_\gamma})}{1-\theta}$ and $\widetilde{d}_2 = 8\big(1 \vee \kappa^2\frac{(c_\gamma+1)}{\|\mathcal{T}_\rho\|}\big)$.

Proof. Fix $\delta \in (0,1)$ and set $\lambda = (nm)^{-p}$ with $p \in (0,1)$. Lemma 2 implies that with probability at least $1 - \frac{\delta}{2}$ the following holds for any $k \in \mathbb{N}$

$$\|(\mathcal{T}_\rho + \lambda I)^{-1/2} N_k\|_H \leq 4(R\kappa^{2r} + \sqrt{M})\left(\kappa + \sqrt{2\sqrt{\nu}c_\gamma}\right)\frac{\log\frac{8}{\delta}}{(nm)^{(1-p\gamma)/2}}.$$

Similarly, Lemma 3 implies that the following holds with probability at least $1 - \frac{\delta}{2}$

$$\begin{aligned}&\|\mathcal{T}_\rho^{1/2}(\mathcal{T}_{\mathbf{x}} + \lambda I)^{-1/2}\|^2 \leq \|\mathcal{T}_{\rho,\lambda}^{1/2}(\mathcal{T}_{\mathbf{x}} + \lambda I)^{-1/2}\|^2 \\ &\leq 48\kappa^2 \min\left(\frac{1}{e(1-p)_+}, \log nm\right)\log\frac{8\kappa^2(c_\gamma+1)}{\delta\|\mathcal{T}_\rho\|}.\end{aligned}$$

Following [27], the series can be bounded as follows

$$\begin{aligned}&\sum_{k=1}^{t-1}\frac{\eta_k}{2\sum_{i=k+1}^{t}\eta_i} + \lambda\sum_{k=1}^{t-1}\eta_k + \|\mathcal{T}_\rho\|^{1/2}(\|\mathcal{T}_\rho\| + \lambda)^{1/2}\eta_t \\ &\leq 2\log(t) + \frac{\lambda\eta t^{1-\theta}}{1-\theta} + \|\mathcal{T}_\rho\|^{1/2}(\|\mathcal{T}_\rho\| + 1)^{1/2}\eta t^{-\theta} \\ &\leq \frac{4\|\mathcal{T}_\rho\|^{1/2}(\|\mathcal{T}_\rho\| + 1)^{1/2}\log(t)}{1-\theta}(1 \vee (\lambda\eta t^{1-\theta}) \vee (\eta t^{-\theta})),\end{aligned}$$

where we used $\lambda = (nm)^{-p} \leq 1$ to get $(\|\mathcal{T}_\rho\| + \lambda)^{1/2} \leq (\|\mathcal{T}_\rho\| + 1)^{1/2}$. Plugging everything into Lemma 1 and using a union bound we obtain that the result holds with probability at least $1 - \frac{\delta}{2} - \frac{\delta}{2} = 1 - \delta$. □

Proposition 3 gives a bound that holds with high probability. We make use of the following lemma to derive a bound in expectation.

Lemma 4. *[7, Appendix Lemma C.1] Let $F : (0,1] \to \mathbb{R}_+$ be a monotone, non-increasing, continuous function and V a non-negative real-valued random variable such that*

$$\mathbf{P}[V > F(t)] \leq t, \quad \forall t \in (0,1].$$

Then we have $\mathbf{E}[V] \leq \int_0^1 F(t)dt$.

The following theorem presents the final bound for the expected value of the Sample Variance term.

Theorem 3. *Let Assumptions 1, 2, 3 hold with $r \geq 1/2$, $p \in (0,1)$ and $\eta_t = \eta t^{-\theta}$ for all $t \in \mathbb{N}$ with $\eta \in (0, \kappa^{-2}]$, $\theta \in [0,1)$. Then for following holds for all $t \in \mathbb{N}$:*

$$\begin{aligned}&\mathbf{E}[\|\mathcal{S}_\rho(\xi_t - \mu_t)\|_\rho^2] \\ &\leq \widetilde{d}_3 \min\left(\frac{1}{e(1-p)_+}, \log nm\right)^2 \frac{\log^2(t)}{(nm)^{(1-p\gamma)}}\left(1 \vee ((nm)^{-p}\eta t^{1-\theta})^2 \vee t^{-2}(\eta t^{1-\theta})^2\right),\end{aligned}$$

with $\widetilde{d}_3 = 64\widetilde{d}_1^2\log^4\widetilde{d}_2$ and with $\widetilde{d}_1$, $\widetilde{d}_2$ defined as in Proposition 3.

Proof. Consider the term $\|\mathcal{S}_\rho(\xi_t - \mu_t)\|_\rho^2$. Utilising the high-probability bound in Proposition 3 as well as Lemma 4, the expectation of the squared norm can be bounded as

$$\begin{aligned}&\mathbf{E}[\|\mathcal{S}_\rho(\xi_t - \mu_t)\|_\rho^2] \\ &\leq \widetilde{d}_1^2 \min\left(\frac{1}{e(1-p)_+}, \log nm\right)^2 \frac{\log^2(t)}{(nm)^{(1-p\gamma)}}\left(1 \vee ((nm)^{-p}\eta t^{1-\theta})^2 \vee t^{-2}(\eta t^{1-\theta})^2\right) \\ &\quad \times \int_0^1 \log^4\frac{\widetilde{d}_2}{\delta}d\delta.\end{aligned}$$

The result follows by using the bound $\int_0^1 \log^4\frac{\widetilde{d}_2}{\delta}d\delta \leq 64\log^4(\widetilde{d}_2)$. □

C.3 Network Error

In this section we develop the bound for the Network Error term. The following lemma shows that the error can be decomposed into terms similar to $\{N_k\}_{k\in\mathbb{N}}$ defined in (7) for the Sample Variance.

Lemma 5. *For all $t \in \mathbb{N}$ we have*

$$\|\mathcal{S}_\rho(\omega_{t+1,v} - \xi_{t+1,v})\|_\rho = \left\| \sum_{k=1}^{t} \eta_k \sum_{w_{t:k}\in V^{t-k+1}} \big(P_{vw_{t:k}} - \frac{1}{n^{t-k+1}}\big)\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t:k+1}}})N_{k,w_k} \right\|_H,$$

where

$$N_{k,v} := (\mathcal{T}_\rho\mu_k - \mathcal{S}_\rho^\star f_\rho) - (\mathcal{T}_{\mathbf{x}_v}\mu_k - \mathcal{S}^\star_{\mathbf{x}_v}\mathbf{y}_v), \quad \forall k \in \mathbb{N},\ v \in V. \tag{8}$$

Proof. For $t \geq 1$ the difference between the iterates $\omega_{t+1,v} - \mu_{t+1}$ can be written as follows

$$\begin{aligned}
\omega_{t+1,v} - \mu_{t+1} &= \sum_{w\in V} P_{vw}\Big(\omega_{t,w} - \mu_t + \eta_t\big\{(\mathcal{T}_\rho\mu_t - \mathcal{S}_\rho^\star f_\rho) - (\mathcal{T}_{\mathbf{x}_w}\omega_{t,w} - \mathcal{S}^\star_{\mathbf{x}_w}\mathbf{y}_w)\big\}\Big) \\
&= \sum_{w\in V} P_{vw}\Big((I - \eta_t\mathcal{T}_{\mathbf{x}_w})(\omega_{t,w} - \mu_t) + \eta_t \underbrace{\big\{(\mathcal{T}_\rho\mu_t - \mathcal{S}_\rho^\star f_\rho) - (\mathcal{T}_{\mathbf{x}_w}\mu_t - \mathcal{S}^\star_{\mathbf{x}_w}\mathbf{y}_w)\big\}}_{N_{t,w}}\Big) \\
&= \sum_{w\in V} P_{vw}\Big((I - \eta_t\mathcal{T}_{\mathbf{x}_w})(\omega_{t,w} - \mu_t) + \eta_t N_{t,w}\Big).
\end{aligned}$$

Unravelling the iterates and using $\omega_1 = \mu_1 = 0$ yield

$$\begin{aligned}
\omega_{t+1,v} - \mu_{t+1} &= \sum_{w_{t:1}\in V^t} P_{vw_{t:1}}\Pi_{t:1}(\mathcal{T}_{\mathbf{x}_{w_{t:1}}})(\omega_1 - \mu_1) + \sum_{k=1}^{t}\eta_k \sum_{w_{t:k}\in V^t} P_{vw_{t:k}}\Pi_{t:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t:k+1}}})N_{k,w_k} \\
&= \sum_{k=1}^{t}\eta_k \sum_{w_{t:k}\in V^{t-k+1}} P_{vw_{t:k}}\Pi_{t:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t:k+1}}})N_{k,w_k}.
\end{aligned}$$

The iterates $\xi_{t+1,v} - \mu_{t+1}$ are similarly written and unravelled using $\xi_{1,v} = 0$:

$$\begin{aligned}
\xi_{t+1,v} - \mu_{t+1} &= \sum_{w\in V}\frac{1}{n}\Big((I - \eta_t\mathcal{T}_{\mathbf{x}_w})(\xi_{t,w} - \mu_t) + \eta_t N_{t,w}\Big) \\
&= \sum_{k=1}^{t}\eta_k \sum_{w_{t:k}\in V^{t-k+1}} \frac{1}{n^{t-k+1}}\Pi_{t:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t:k+1}}})N_{k,w_k}.
\end{aligned}$$

The deviation $\omega_{t+1} - \xi_{t+1,v}$ can then be written as follows

$$\omega_{t+1,v} - \xi_{t+1,v} = \sum_{k=1}^{t}\eta_k \sum_{w_{t:k}\in V^{t-k+1}} \left(P_{vw_{t:k}} - \frac{1}{n^{t-k+1}}\right)\Pi_{t:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t:k+1}}})N_{k,w_k}.$$

Applying $\mathcal{S}_\rho$, taking norm $\|\cdot\|_\rho$ on both sides and using the isometry property yields the result. □

For $v, w \in V$ and $k \geq 1$, we want to exploit that the random variables $N_{k,v}$ and $N_{k,w}$ have zero mean, $\mathbf{E}[N_{k,v}] = 0$, and are independent for $v \neq w$. To do so we add and subtract $\Pi_{t:k+1}(\mathcal{T}_\rho)$ inside the norm so the following upper bound can be formed:

$$\begin{aligned}
&\|\mathcal{S}_\rho(\omega_{t+1,v} - \xi_{t+1,v})\|_\rho^2 \qquad (9)\\
&\leq 2\underbrace{\left\| \sum_{k=1}^{t}\eta_k \sum_{w_{t:k}\in V^{t-k+1}} \big(P_{vw_{t:k}} - \frac{1}{n^{t-k+1}}\big)\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)N_{k,w_k} \right\|_H^2}_{(\textbf{Population Covariance Error})^2} \\
&+ 2\underbrace{\left\| \sum_{k=1}^{t}\eta_k \sum_{w_{t:k}\in V^{t-k+1}} \big(P_{vw_{t:k}} - \frac{1}{n^{t-k+1}}\big)\mathcal{T}_\rho^{1/2}\big(\Pi_{t:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t:k+1}}}) - \Pi_{t:k+1}(\mathcal{T}_\rho)\big)N_{k,w_k} \right\|_H^2}_{(\textbf{Residual Empirical Covariance Error})^2}.
\end{aligned}$$

The **Population Covariance Error** (**Pop. Cov. Error**) will be controlled by using the independence of the terms $\{N_{k,w}\}_{w\in V}$. The **Residual Empirical Covariance Error** (**Resid. Emp. Cov. Error**) will be analysed by decomposing it into terms that concentrate to zero sufficiently quickly.

The following lemma, similar to Lemma 2 for the sample variance, gives concentration rates for the quantities held by the individual agents.

Lemma 6. *Fix $v \in V$. Let Assumptions 1, 2, 3 hold with $r \geq 1/2$ and $\{N_{s,v}\}_{s\in\mathbb{N}}$ be defined as in* (8)*. For any $\lambda > 0$, with probability at least $1-\delta$, the following holds for all $k \in \mathbb{N}$:*

$$\|(\mathcal{T}_\rho + \lambda I)^{-1/2} N_{k,v}\|_H \leq 4(R\kappa^{2r} + \sqrt{M})\left(\frac{\kappa}{m\sqrt{\lambda}} + \frac{\sqrt{2\sqrt{\nu}c_\gamma}}{\sqrt{m\lambda^\gamma}}\right)\log\frac{4}{\delta}. \tag{10}$$

Let $\|\cdot\|_{HS}$ denote the Hilbert-Schmidt norm of a bounded operator from H to H. The following holds with probability at least $1-\delta$:

$$\|(\mathcal{T}_\rho + \lambda I)^{-1/2}(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_v})\|_{HS} \leq 2\kappa\left(\frac{2\kappa}{m\sqrt{\lambda}} + \frac{\sqrt{c_\gamma}}{\sqrt{m\lambda^\gamma}}\right)\log\frac{4}{\delta}. \tag{11}$$

Proof. Both inequalities arise from concentration results for random variables in Hilbert spaces used in [12] and based on results in [36]. Inequalities (10,11) come directly from [27, Lemma 18], where in particular (11) was used to prove (10). □

We now move on to establish bounds for the **Population Covariance Error** term and the **Residual Empirical Covariance Error** term within the following two sections, Section C.3.1 and Section C.3.2, respectively. Section C.3.3 then brings together the previously developed results to establish a bound for the Network Error term.

We will need the following lemma, taken from [27, Lemma 15], which itself follows [58, 49].

Lemma 7. *Let $\mathcal{L}$ be a compact, positive operator on a separable Hilbert Space H. Assume that $\eta\|\mathcal{L}\| \leq 1$. For $t \in \mathbb{N}$, $a > 0$ and any non-negative integer $k \leq t-1$ we have*

$$\|\Pi_{t:k+1}(\mathcal{L})\mathcal{L}^a\| \leq \left(\frac{a}{e\sum_{j=k+1}^t \eta_j}\right)^a.$$

Proof. The proof in [27, Lemma 15] considers this result with $a = r$. The proof for more general $a > 0$ follows the same steps. □

C.3.1 Analysis of Population Covariance Error

In this section we develop a bound for the **Population Covariance Error** term in (9). The final result is presented in Lemma 8.

The following proposition bounds the expectation of (**Population Covariance Error**)^2 by a series involving the products of (deterministic) operators $\{\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)\}$, as a function of the step size, the largest eigenvalue in absolute value of the gossip matrix P, and the random variables $\{N_{k,w}\}$.

Proposition 4. *For any $t \in \mathbb{N}$ and $v \in V$ we have*

$$\mathbf{E}\left[\left\|\sum_{k=1}^t \eta_k \sum_{w_{t:k}\in V^{t-k+1}}\left(P_{vw_{t:k}} - \frac{1}{n^{t-k+1}}\right)\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)N_{k,w_k}\right\|_H^2\right]$$
$$\leq \mathbf{E}\left[\left(\sum_{k=1}^t \sigma_2^{t-k+1}\eta_k\|\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)N_{k,v}\|_H\right)^2\right].$$

Proof. Fix $t \in \mathbb{N}$ and $v \in V$. Let us introduce the notation $\Delta(w_{t:k}) := \big(P_{vw_{t:k}} - \frac{1}{n^{t-k+1}}\big)$. Expanding the square and taking the expectation we get

$$\begin{aligned}
&\mathbf{E}\Big[\Big\|\sum_{k=1}^{t}\eta_k \sum_{w_{t:k}\in V^{t-k+1}}\Big(P_{vw_{t:k}} - \frac{1}{n^{t-k+1}}\Big)\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)N_{k,w_k}\Big\|_H^2\Big] \\
&= \sum_{k,k'=1}^{t}\eta_k\eta_{k'}\sum_{\substack{w_{t:k}\in V^{t-k+1}\\ w'_{t:k'}\in V^{t-k'+1}}}\Delta(w_{t:k})\Delta(w'_{t:k'})\mathbf{E}\langle\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)N_{k,w_k}, \mathcal{T}_\rho^{1/2}\Pi_{t:k'+1}(\mathcal{T}_\rho)N_{k',w'_{k'}}\rangle_H \\
&= \sum_{k,k'=1}^{t}\eta_k\eta_{k'}\mathbf{E}\langle\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)N_{k,v}, \mathcal{T}_\rho^{1/2}\Pi_{t:k'+1}(\mathcal{T}_\rho)N_{k',v}\rangle_H \sum_{\substack{w_{t:k}\in V^{t-k+1}\\ w'_{t:k'}\in V^{t-k'+1}\\ w_k=w'_{k'}}}\Delta(w_{t:k})\Delta(w'_{t:k'}).
\end{aligned}$$

The last identity follows from the fact that the samples held by agents are independent and identically distributed. As the agents' datasets are independent, the inner products are zero for $k, k' \in \{1,\dots,t\}$ whenever the final elements of the paths $w_{t:k}$ and $w'_{t:k'}$ do not coincide, i.e.

$$\mathbf{E}\langle\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)N_{k,w_k}, \mathcal{T}_\rho^{1/2}\Pi_{t:k'+1}(\mathcal{T}_\rho)N_{k',w'_{k'}}\rangle_H = 0 \text{ if } w_k \neq w'_{k'}.$$

As the agents' datasets are identically distributed, the expectation of the inner products can be taken outside the sum over the paths. The sum over all pairs of paths that intersect at the final node can be simplified as follows:

$$\begin{aligned}
&\sum_{\substack{w_{t:k}\in V^{t-k+1}\\ w'_{t:k'}\in V^{t-k'+1}\\ w_k=w'_{k'}}}\Delta(w_{t:k})\Delta(w'_{t:k'}) \\
&= \sum_{\substack{w_k,w'_{k'}\in V\\ w_k=w'_{k'}}}\sum_{w_{t:k+1}\in V^{t-k}}\sum_{w'_{t:k+1}\in V^{t-k'}}\Big(P_{vw_{t:k}} - \frac{1}{n^{t-k+1}}\Big)\Big(P_{vw'_{t:k'}} - \frac{1}{n^{t-k'+1}}\Big) \\
&= \sum_{w\in V}\Big((P^{t-k+1})_{vw} - \frac{1}{n}\Big)\Big((P^{t-k'+1})_{vw} - \frac{1}{n}\Big).
\end{aligned}$$

For each $v \in V$ let $e_v \in \mathbb{R}^n$ denote the vector of all zeros but a 1 in the place aligned with agent v. The summation can be further simplified by utilising the assumption that P is symmetric and doubly-stochastic, i.e. $P^\top = P$ and $P\mathbf{1} = \mathbf{1}$. By the eigendecomposition of the gossip matrix P, recall Section 2.3, for any $s > 0$ we have $(P^s)_{vv} = \sum_{l=1}^n \lambda_l^s u_{l,v}^2 = \frac{1}{n} + \sum_{l=2}^n \lambda_l^s u_{l,v}^2$. This yields the bound $|(P^s)_{vv} - \frac{1}{n}| = |\sum_{l=2}^n \lambda_l^s u_{l,v}^2| \leq \sigma_2^s \sum_{l=2}^n u_{l,v}^2 \leq \sigma_2^s$ where $\sigma_2 := \max\{|\lambda_2|, |\lambda_n|\}$ is the second largest eigenvalue in absolute value. Bringing everything together, the expected norm of $(\textbf{Pop. Cov. Error})^2$ can be written and bounded as follows:

$$\begin{aligned}
&\mathbf{E}\Big[\Big\|\sum_{k=1}^{t}\eta_k \sum_{w_{t:k}\in V^{t-k+1}}\Big(P_{vw_{t:k}} - \frac{1}{n^{t-k+1}}\Big)\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)N_{k,w_k}\Big\|_H^2\Big] \\
&= \sum_{k,k'=1}^{t}\eta_k\eta_{k'}\mathbf{E}\langle\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)N_{k,v}, \mathcal{T}_\rho^{1/2}\Pi_{t:k'+1}(\mathcal{T}_\rho)N_{k',v}\rangle_H\Big(P_{vv}^{2t-k-k'+2} - \frac{1}{n}\Big) \\
&\leq \sum_{k,k'=1}^{t}\eta_k\eta_{k'}\mathbf{E}|\langle\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(T_\rho)N_{k,v}, \mathcal{T}_\rho^{1/2}\Pi_{t:k'+1}(\mathcal{T}_\rho)N_{k',v}\rangle_H|\,\Big|\Big(P_{vv}^{2t-k-k'+2} - \frac{1}{n}\Big)\Big| \\
&\leq \sum_{k,k'=1}^{t}\eta_k\eta_{k'}\mathbf{E}\big[\|\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)N_{k,v}\|_H\|\mathcal{T}_\rho^{1/2}\Pi_{t:k'+1}(\mathcal{T}_\rho)N_{k',v}\|_H\big]\sigma_2^{2t-k-k'+2} \\
&= \mathbf{E}\Big[\Big(\sum_{k=1}^{t}\eta_k\sigma_2^{t-k+1}\|\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)N_{k,v}\|_H\Big)^2\Big],
\end{aligned}$$

where we used Jensen's inequality and the Cauchy-Schwarz inequality. □

The following lemma presents the final bound for the **Population Covariance Error**. This result is established by utilising the series bound in Proposition 4 to split the error into well-mixed and poorly-mixed terms, i.e. for k such that $t-k \gtrsim 1/(1-\sigma_2)$ and $t-k \lesssim 1/(1-\sigma_2)$. The well-mixed terms are controlled using that σ_2^{t-k+1} is small. The poorly-mixed terms (there are $\sim 1/(1-\sigma_2)$ of them) are controlled using both the concentration of the error terms $\{N_{k,w}\}_{k\geq 1, w\in V}$ as well as the contractive nature of the gradient updates, i.e. the operator norm of $\{\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)\}$ in Lemma 7. The contractive terms arising from the gradient updates are decreasing in the step size: larger steps achieve a faster contraction. However, each term within the Network Error series is scaled by the step size $\{\eta_k\}_{k\geq 1}$, i.e. the Network Error takes the form $\sum_{k=1}^t \sigma_2^{t-k+1}\eta_k[\cdots]$ where $[\cdots]$ indicates the right most terms. To exploit this trade-off we introduce two free parameters $\alpha \in [0, 1/2]$ and $\gamma' \in [1, \gamma]$, which describe the degree to which the contraction is utilised. Specifically, $\alpha = 0$ and $\gamma' = \gamma$ is the large step regime and, $\alpha = 1/2$ and $\gamma' = 1$ is the small step regime.

Lemma 8. *Let Assumptions 1, 2, 3 hold with $r \geq 1/2$, $\eta_t = \eta t^{-\theta}$ for $t \in \mathbb{N}$ with $\eta\kappa^2 \leq 1$ and $\theta \in [0, 1)$. The following holds for any $v \in V$, $t/2 \geq \lceil \frac{(1+r)\log(t)}{1-\sigma_2} \rceil =: t^\star$, $\alpha \in [0, 1/2]$ and $\gamma' \in [1, \gamma]$:*

$$\mathbf{E}\Big[\Big\| \sum_{k=1}^t \eta_k \sum_{w_{t:k}\in V^{t-k+1}} \Big(P_{vw_{t:k}} - \frac{1}{n^{t-k+1}}\Big)\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)N_{k,w_k}\Big\|_H^2\Big]$$
$$\leq \frac{\widetilde{a}\log^2(4n)\log^2(t^\star)}{m}\Big(\eta^2 t^{-2r} \vee (m^{-1}(\eta t^\star)^{1+2\alpha}) \vee (\eta t^\star)^{\gamma'+2\alpha}\Big),$$

where
$\widetilde{a} = \frac{1152(R\kappa^{2r}+\sqrt{M})^2(\kappa+\sqrt{2\sqrt{\nu}c_{\gamma'}})^2(\|\mathcal{T}_\rho\|\vee 1)^2}{\|\mathcal{T}_\rho\|\wedge\|\mathcal{T}_\rho\|^{\gamma'}}\Big[6\Big(\frac{\|\mathcal{T}_\rho^\alpha\|t^{-\alpha\theta}}{\alpha}\vee\frac{t^{-(\alpha+1/2)\theta}\|\mathcal{T}_\rho^\alpha\|}{1/2+\alpha}\vee t^{-\theta}\|\mathcal{T}_\rho\|\Big)\mathbb{1}_{\{\alpha\neq 0\}}+10\Big]^2.$

Proof. Consider the bound of **Population Covariance Error** in Proposition 4. Let $\|\mathcal{T}_\rho\| \geq \lambda \geq 0$, $\widetilde{\lambda} \geq 0$ and for $c > 0$ introduce the cutoff $t^\star = \lceil \frac{c\log(t)}{1-\sigma_2} \rceil$. For $k = 1, \ldots, t$ and $v \in V$ we have

$$\|\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)N_{k,v}\|_H \leq \|\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)\mathcal{T}_{\rho,\lambda}^{1/2}\|\|\mathcal{T}_{\rho,\lambda}^{-1/2}N_{k,v}\|_H$$
$$\leq \|\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)\mathcal{T}_{\rho,\lambda}^{1/2}\| \max_{k=1,\ldots,t}\Big\{\|\mathcal{T}_{\rho,\lambda}^{-1/2}N_{k,v}\|_H\Big\},$$

and similarly for $\widetilde{\lambda}$. Let us split the summation at $k \leq t - t^\star - 1$ and $k \geq t - t^\star$ using the bound above to obtain

$$\Big(\sum_{k=1}^t \sigma_2^{t-k+1}\eta_k\|\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)N_{k,v}\|_H\Big)^2$$
$$\leq 2\underbrace{\Big(\sum_{k=1}^{t-t^\star-1}\sigma_2^{t-k+1}\eta_k\|\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)\mathcal{T}_{\rho,\lambda}^{1/2}\|\Big)^2}_{\textbf{Well-Mixed Network Error}} \max_{k=1,\ldots,t}\Big\{\|\mathcal{T}_{\rho,\lambda}^{-1/2}N_{k,v}\|_H^2\Big\}$$
$$+ 2\underbrace{\Big(\sum_{k=t-t^\star}^{t}\sigma_2^{t-k+1}\eta_k\|\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\|\Big)^2}_{\textbf{Poorly-Mixed Network Error}} \max_{k=1,\ldots,t}\Big\{\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{-1/2}N_{k,v}\|_H^2\Big\}.$$

The **Well-Mixed Network Error** is controlled through σ_2^{t-k+1} being small for $k \leq t - t^\star$. From $\|\Pi_{t:k+1}(\mathcal{T}_\rho)\| \leq 1$ and $\lambda \leq \|\mathcal{T}_\rho\|$ we have $\|\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)\mathcal{T}_{\rho,\lambda}^{1/2}\|_H \leq 2\|\mathcal{T}_\rho\|$, and from $1/\log(1/\sigma_2) \leq 1/(1-\sigma_2)$ we have $t^\star \geq \frac{c\log(t)}{-\log(\sigma_2)}$. These two facts allow the **Well-Mixed Network Error** to be bounded as follows:

$$\textbf{Well-Mixed Network Error} \leq 2\|\mathcal{T}_\rho\|\eta\sum_{k=1}^{t-t^\star}\sigma_2^{t-k+1}k^{-\theta} \leq 2\eta\|\mathcal{T}_\rho\|\sum_{k=1}^{t-t^\star}\sigma_2^{\frac{c\log(t)}{-\log(\sigma_2)}} \leq 2\eta\|\mathcal{T}_\rho\|t^{1-c}.$$

For the **Poorly-Mixed Network Error** let us consider the two cases $\alpha \in (0, 1/2]$ and $\alpha = 0$ separately. Consider $\alpha \in (0, 1/2]$ first. Using Lemma 7[4] we have, for $t - 1 \geq k \geq 1$,

$$\begin{aligned}\|\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\| &\leq \|\mathcal{T}_\rho\Pi_{t:k+1}(\mathcal{T}_\rho)\| + \sqrt{\widetilde{\lambda}}\|\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)\| \\ &\leq \|\mathcal{T}_\rho^\alpha\|\|\mathcal{T}_\rho^{1-\alpha}\Pi_{t:k+1}(\mathcal{T}_\rho)\| + \sqrt{\widetilde{\lambda}}\|\mathcal{T}_\rho^\alpha\|\|\mathcal{T}_\rho^{1/2-\alpha}\Pi_{t:k+1}(\mathcal{T}_\rho)\| \\ &\leq \|\mathcal{T}_\rho^\alpha\|\bigg(\frac{1-\alpha}{e\sum_{j=k+1}^t \eta_j}\bigg)^{1-\alpha} + \sqrt{\widetilde{\lambda}}\|\mathcal{T}_\rho^\alpha\|\bigg(\frac{1/2-\alpha}{e\sum_{j=k+1}^t \eta_j}\bigg)^{1/2-\alpha}.\end{aligned}$$

When plugging the above into the **Poorly-Mixed Network Error**, summations of the form $\sum_{k=t-t^\star}^{t-1} \frac{\eta_k}{(\sum_{j=k+1}^t \eta_j)^\beta}$ appear for $\beta = 1 - \alpha$ and $\beta = 1/2 - \alpha$. To bound these consider the following for $\beta \in [0, 1)$ and $t \geq 2t^\star$:

$$\begin{aligned}\sum_{k=t-t^\star}^{t-1} \frac{\eta_k}{\big(\sum_{j=k+1}^t \eta_j\big)^\beta} &= \eta^{1-\beta}\sum_{k=t-t^\star}^{t-1} \frac{k^{-\theta}}{\big(\sum_{j=k+1}^t j^{-\theta}\big)^\beta} \\ &\leq \eta^{1-\beta}t^{\theta\beta}\sum_{k=t-t^\star}^{t-1} \frac{k^{-\theta}}{\big(t-k\big)^\beta} \\ &\leq \frac{\eta^{1-\beta}t^{\theta\beta}}{(t-t^\star)^\theta}\sum_{k=t-t^\star}^{t-1} \frac{1}{\big(t-k\big)^\beta} \\ &= \frac{\eta^{1-\beta}t^{\theta\beta}}{(t-t^\star)^\theta}\sum_{k=1}^{t^\star} \frac{1}{k^\beta} \\ &\leq 2\eta^{1-\beta}t^{\theta(\beta-1)}\frac{(t^\star)^{1-\beta}}{1-\beta},\end{aligned}$$

where the last inequality follows from an integral bound as well as using that $\frac{t^{\theta\beta}}{(t-t^\star)^\theta} = \frac{t^{\theta(\beta-1)}}{(1-\frac{t^\star}{t})^\theta} \leq 2t^{\theta(\beta-1)}$ from $t \geq 2t^\star$. Splitting the summation at $k = t$, plugging the above two bounds into the **Poorly-Mixed Network Error** term and using $(\eta t^\star)^\alpha \geq \eta$ from $\eta \leq \kappa^{-2} \leq 1$ yields a bound for $\alpha \in (0, 1/2]$:

Poorly-Mixed Network Error

$$\begin{aligned}&\leq \frac{2\|\mathcal{T}_\rho^\alpha\|t^{-\alpha\theta}}{\alpha}(\eta t^\star)^\alpha + \frac{2t^{-(\alpha+1/2)\theta}\|\mathcal{T}_\rho^\alpha\|}{1/2+\alpha}\sqrt{\widetilde{\lambda}}(\eta t^\star)^{1/2+\alpha} + \sqrt{2}\eta t^{-\theta}\|\mathcal{T}_\rho\| \\ &\leq 6\bigg(\frac{\|\mathcal{T}_\rho^\alpha\|t^{-\alpha\theta}}{\alpha} \vee \frac{t^{-(\alpha+1/2)\theta}\|\mathcal{T}_\rho^\alpha\|}{1/2+\alpha} \vee t^{-\theta}\|\mathcal{T}_\rho\|\bigg)((\eta t^\star)^\alpha \vee \sqrt{\widetilde{\lambda}}(\eta t^\star)^{1/2+\alpha}).\end{aligned}$$

Now consider the case $\alpha = 0$. The summation for $\beta = 1$ in this case is bounded following the previous steps

$$\sum_{k=t-t^\star}^{t-1} \frac{\eta_k}{\big(\sum_{j=k+1}^t \eta_j\big)} \leq \frac{t^\theta}{(t-t^\star)^\theta}\sum_{k=t-t^\star}^{t-1} \frac{1}{(t-k)} \leq 2^{1+\theta}\log(t^\star),$$

leading to the **Poorly-Mixed Network Error** bounded as for $\alpha = 0$ from $\eta\|\mathcal{T}_\rho\| \leq 1$:

$$\begin{aligned}\textbf{Poorly-Mixed Network Error} &\leq 2^{1+\theta}\log(t^\star) + 4t^{-\theta/2}\sqrt{\widetilde{\lambda}}(\eta t^\star)^{1/2} + \sqrt{2}\eta t^{-\theta}\|\mathcal{T}_\rho\| \\ &\leq 10\log(t^\star)(1 \vee (\sqrt{\widetilde{\lambda}}(\eta t^\star)^{1/2})).\end{aligned}$$

[4] The operator norm can be bounded $\|\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)\mathcal{T}_{\rho,\lambda}^{1/2}\| \leq \sup_{x\in(0,\kappa^2)}\big\{x^{1/2}(x+\lambda)^{1/2}\prod_{\ell=k+1}^t(1-\eta_\ell x)\big\} \leq \sup_{x\in(0,\kappa^2)}\big\{x\prod_{\ell=k+1}^t(1-\eta_\ell x)\big\} + \sqrt{\lambda}\sup_{x\in(0,\kappa^2)}\big\{x^{1/2}\prod_{\ell=k+1}^t(1-\eta_\ell x)\big\}$. Using techniques used to prove [27, Lemma 15], these terms can be bounded as shown.

Combining the two bounds for $\alpha = 0$ and $\alpha \in (0, 1/2]$ gives

Poorly-Mixed Network Error

$$\leq \log(t^\star)\left[6\left(\frac{\|\mathcal{T}_\rho^\alpha\| t^{-\alpha\theta}}{\alpha} \vee \frac{t^{-(\alpha+1/2)\theta}\|\mathcal{T}_\rho^\alpha\|}{1/2+\alpha} \vee t^{-\theta}\|\mathcal{T}_\rho\|\right)\mathbb{1}_{\{\alpha\neq 0\}} + 10\right]((\eta t^\star)^\alpha \vee \sqrt{\widetilde{\lambda}}(\eta t^\star)^{1/2+\alpha}).$$

We now consider the terms $\max_{k=1,\ldots,t}\{\|\mathcal{T}_{\rho,\lambda}^{-1/2} N_{k,v}\|_H^2\}$ for both λ and $\widetilde{\lambda}$. We use the high-probability bounds of Lemma 6 to uniformly control $\|\mathcal{T}_{\rho,\lambda}^{-1/2} N_{k,v}\|_H^2$ for all $k = 1, \ldots, t$ and $v \in V$. For $w \in V$, let $\delta_w = \frac{\delta}{n}$. With probability at least $1 - \delta_w$ the following holds for all $k = 1, \ldots, t$ and $\gamma' \in [1, \gamma]$:

$$\|\mathcal{T}_{\rho,\lambda}^{-1/2} N_{k,w}\|_H^2 \leq 16(R\kappa^{2r} + \sqrt{M})^2\left(\frac{\kappa}{m\sqrt{\lambda}} + \frac{\sqrt{2\sqrt{\nu}c_{\gamma'}}}{\sqrt{m\lambda^{\gamma'}}}\right)^2 \log^2\frac{4n}{\delta}.$$

We note that if the capacity assumption holds for γ, then it also holds for all $\gamma' \in [1, \gamma]$. Applying a union bound, we get that the above holds with probability at least $1 - \sum_{v\in V}\delta_v = 1 - \delta$ for all $w \in V$ and $k = 1, \ldots, t$. Using Lemma 4, the expectation of the maximum can be bounded for any $v \in V$ and $\gamma' \in [1, \gamma]$ as follows:

$$\begin{aligned}
&\mathbf{E}\big[\max_{k=1,\ldots,t}\big\{\|\mathcal{T}_{\rho,\lambda}^{-1/2} N_{k,v}\|_H^2\big\}\big] \\
&\leq 16(R\kappa^{2r} + \sqrt{M})^2\left(\frac{\kappa}{m\sqrt{\lambda}} + \frac{\sqrt{2\sqrt{\nu}c_{\gamma'}}}{\sqrt{m\lambda^{\gamma'}}}\right)^2 \int_0^1 \log^2\frac{4n}{\delta} d\delta \\
&\leq 96(R\kappa^{2r} + \sqrt{M})^2\left(\frac{\kappa}{m\sqrt{\lambda}} + \frac{\sqrt{2\sqrt{\nu}c_{\gamma'}}}{\sqrt{m\lambda^{\gamma'}}}\right)^2 \log^2 4n,
\end{aligned}$$

where we used $\int_0^1 \log^2\frac{4n}{\delta} d\delta \leq 6\log^2 4n$.

Bringing together the bounds for the **Poorly-** and **Well-Mixed Network Error** with the above bound for the quantity $\mathbf{E}\big[\max_{k=1,\ldots,t}\big\{\|\mathcal{T}_{\rho,\lambda}^{-1/2} N_{k,v}\|_H^2\big\}\big]$ yields

$$\begin{aligned}
&\mathbf{E}\left[\left(\sum_{k=1}^t \sigma_2^{t-k+1}\eta_k \|\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)N_{k,v}\|_H\right)^2\right] \\
&\leq 96\log^2(4n)\log^2(t^\star)(R\kappa^{2r} + \sqrt{M})^2 \\
&\quad\times\Bigg(8\|\mathcal{T}_\rho\|^2\Big(\frac{\kappa}{m\sqrt{\lambda}} + \frac{\sqrt{2\sqrt{\nu}c_\gamma}}{\sqrt{m\lambda^\gamma}}\Big)^2\eta^2 t^{2(1-c)} \\
&\qquad + 2\Big[6\Big(\frac{\|\mathcal{T}_\rho^\alpha\| t^{-\alpha\theta}}{\alpha} \vee \frac{t^{-(\alpha+1/2)\theta}\|\mathcal{T}_\rho^\alpha\|}{1/2+\alpha} \vee t^{-\theta}\|\mathcal{T}_\rho\|\Big)\mathbb{1}_{\{\alpha\neq 0\}} + 10\Big]^2\Big(\frac{\kappa}{m\sqrt{\widetilde{\lambda}}} + \frac{\sqrt{2\sqrt{\nu}c_{\gamma'}}}{\sqrt{m\widetilde{\lambda}^{\gamma'}}}\Big)^2 \\
&\qquad\times\Big((\eta t^\star)^{2\alpha} \vee \widetilde{\lambda}(\eta t^\star)^{1+2\alpha}\Big)\Bigg).
\end{aligned}$$

Let $\lambda = \|\mathcal{T}_\rho\|$ and $\widetilde{\lambda} = \frac{\|\mathcal{T}_\rho\|}{\eta t^\star}$. The bound

$$\begin{aligned}
\frac{1}{m\sqrt{\widetilde{\lambda}}} + \frac{1}{\sqrt{m\widetilde{\lambda}^{\gamma'}}} &\leq \frac{2}{\sqrt{m}}\left(\frac{1}{\sqrt{m\|\mathcal{T}_\rho\|(\eta t^\star)^{-1}}} \vee \frac{1}{\|\mathcal{T}_\rho\|^{\gamma'/2}(\eta t^\star)^{-\gamma'/2}}\right) \\
&\leq \frac{2}{\sqrt{m(\|\mathcal{T}_\rho\| \wedge \|\mathcal{T}_\rho\|^{\gamma'})}}\Big(\sqrt{\eta t^\star/m} \vee (\eta t^\star)^{\gamma'/2}\Big)
\end{aligned}$$

allows the expected squared series to be bounded as follows:

$$\begin{aligned}
&\mathbf{E}\left[\left(\sum_{k=1}^t \sigma_2^{t-k+1}\eta_k \|\mathcal{T}_\rho^{1/2}\Pi_{t:k+1}(\mathcal{T}_\rho)N_{k,v}\|_H\right)^2\right] \\
&\leq \frac{\widetilde{a}\log^2(4n)\log^2(t^\star)}{m}\Big((\eta t^{1-c})^2 \vee (m^{-1}(\eta t^\star)^{1+2\alpha}) \vee (\eta t^\star)^{\gamma'+2\alpha}\Big)
\end{aligned}$$

where
$\widetilde{a} = \frac{1152(R\kappa^{2r}+\sqrt{M})^2(\kappa+\sqrt{2\sqrt{\nu}c_{\gamma'}})^2(\|\mathcal{T}_\rho\|\vee 1)^2}{\|\mathcal{T}_\rho\|\wedge\|\mathcal{T}_\rho\|^{\gamma'}}\Big[6\Big(\frac{\|\mathcal{T}_\rho^\alpha\|t^{-\alpha\theta}}{\alpha}\vee\frac{t^{-(\alpha+1/2)\theta}\|\mathcal{T}_\rho^\alpha\|}{1/2+\alpha}\vee t^{-\theta}\|\mathcal{T}_\rho\|\Big)\mathbb{1}_{\{\alpha\neq 0\}}+10\Big]^2$.
The choice $c = 1 + r$ yields the final result. □

C.3.2 Analysis of Residual Empirical Covariance Error

In this section we develop a bound for the **Residual Empirical Covariance Error** term in (9). The final result is presented in Lemma 9.

The following proposition writes the **Residual Empirical Covariance Error** in terms of a series of quantities that will be later controlled.

Proposition 5. *Let $t \geq k+1$. For any $w_{t:k+1} \in V^{t-k}$ we have*

$$\Pi_{t:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t:k+1}}}) = \Pi_{t:k+1}(\mathcal{T}_\rho) + \sum_{j=k+1}^{t} \eta_j \Pi_{t:j+1}(\mathcal{T}_\rho)(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_{w_j}})\Pi_{j-1:k+1}(\mathcal{T}_{\mathbf{x}_{w_{j-1:k+1}}}).$$

Proof. Adding and subtracting $(I-\eta_t\mathcal{T}_\rho)\Pi_{t-1:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t-1:k+1}}})$ and unravelling yields the following:

$$\begin{aligned}
&\Pi_{t:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t:k+1}}}) - \Pi_{t:k+1}(\mathcal{T}_\rho) \\
&= (I-\eta_t\mathcal{T}_{\mathbf{x}_{w_t}})\Pi_{t-1:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t-1:k+1}}}) - (I-\eta_t\mathcal{T}_\rho)\Pi_{t-1:k+1}(\mathcal{T}_\rho) \\
&= (I-\eta_t\mathcal{T}_{\mathbf{x}_{w_t}})\Pi_{t-1:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t-1:k+1}}}) - (I-\eta_t\mathcal{T}_\rho)\Pi_{t-1:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t-1:k+1}}}) \\
&\quad + (I-\eta_t\mathcal{T}_\rho)\Pi_{t-1:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t-1:k+1}}}) - (I-\eta_t\mathcal{T}_\rho)\Pi_{t-1:k+1}(\mathcal{T}_\rho) \\
&= \eta_t(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_{w_t}})\Pi_{t-1:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t-1:k+1}}}) + (I-\eta_t\mathcal{T}_\rho)\big[\Pi_{t-1:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t-1:k+1}}}) - \Pi_{t-1:k+1}(\mathcal{T}_\rho)\big] \\
&= \sum_{j=k+1}^{t} \eta_j \Pi_{t:j+1}(\mathcal{T}_\rho)(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_{w_j}})\Pi_{j-1:k+1}(\mathcal{T}_{\mathbf{x}_{w_{j-1:k+1}}}).
\end{aligned}$$

□

Applying Proposition 5 to the **Residual Empirical Covariance Error** term, using the triangle equality, yields

$$\begin{aligned}
&\Big\|\sum_{k=1}^{t}\eta_k \sum_{w_{t:k}\in V^{t-k+1}} \Delta(w_{t:k})\mathcal{T}_\rho^{1/2}\big(\Pi_{t:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t:k+1}}}) - \Pi_{t:k+1}(\mathcal{T}_\rho)\big)N_{k,w_k}\Big\|_H \\
&\leq \sum_{k=1}^{t-1}\eta_k \sum_{w_{t:k}\in V^{t-k+1}} |\Delta(w_{t:k})| \sum_{j=k+1}^{t}\eta_j \\
&\quad \times \|\mathcal{T}_\rho^{1/2}\Pi_{t:j+1}(\mathcal{T}_\rho)(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_{w_j}})\Pi_{j-1:k+1}(\mathcal{T}_{\mathbf{x}_{w_{j-1:k+1}}})N_{k,w_k}\|_H, \qquad (12)
\end{aligned}$$

where the quantity is zero in the case $k = t$. For $j \in \{2,\dots,t-1\}$ the above includes the quantity $\Pi_{t:j+1}(\mathcal{T}_\rho)$. This can be interpreted in a similar manner to the filter function associated for gradient descent, see for instance [26, Example 2]. In this context it is used to control the growth of the above error term, which is absent in the case $j = t$. This yields the following proposition.

Proposition 6. *Let Assumptions 1, 2, 3 hold with $r \geq 1/2$ and $\eta_t = \eta t^{-\theta}$ for $t \in \mathbb{N}$ with $\eta\kappa^2 \leq 1$, $\theta \in (0,1)$. Fix $\lambda, \widetilde{\lambda} > 0$ and $\delta \in (0,1)$. With probability at least $1-\delta$ the following hold: for any $t-1 \geq j \geq k+1$ and path $w_{t:k} \in V^{t-k+1}$ we have*

$$\begin{aligned}
&\|\mathcal{T}_\rho^{1/2}\Pi_{t:j+1}(\mathcal{T}_\rho)(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_{w_j}})\Pi_{j-1:k+1}(\mathcal{T}_{\mathbf{x}_{w_{j-1:k+1}}})N_{k,w_k}\|_H \\
&\leq 2\kappa\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\|\Big(\frac{1}{\sum_{i=j+1}^{t}\eta_i} + \Big(\frac{\lambda}{\sum_{i=j+1}^{t}\eta_i}\Big)^{1/2}\Big)\Big(\frac{2\kappa}{m\sqrt{\lambda}} + \frac{\sqrt{c_\gamma}}{\sqrt{m\lambda^\gamma}}\Big)\log\Big(\frac{4n}{\delta}\Big) \\
&\quad \times \max_{w\in V}\big\{\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{-1/2}N_{k,w}\|_H\big\}, \qquad (13)
\end{aligned}$$

for any $t-1 \geq k \geq 1$ *and nodes* $w_t, w_k \in V$

$$\begin{aligned}&\|\mathcal{T}_\rho^{1/2}(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_{w_t}})N_{k,w_k}\|_H \\ &\leq 2\kappa\|\mathcal{T}_\rho^{1/2}\mathcal{T}_{\rho,\lambda}^{1/2}\|\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\|\left(\frac{2\kappa}{m\sqrt{\lambda}} + \frac{\sqrt{c_\gamma}}{\sqrt{m\lambda^\gamma}}\right)\log\frac{4n}{\delta}\max_{w\in V}\left\{\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{-1/2}N_{k,w}\|_H\right\}. \end{aligned} \tag{14}$$

Proof. Fix $t-1 \geq j \geq k+1$ and $w_{t:k} \in V^{t-k+1}$. Begin by proving (13). Expanding the norm,

$$\begin{aligned}&\|\mathcal{T}_\rho^{1/2}\Pi_{t:j+1}(\mathcal{T}_\rho)(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_{w_j}})\Pi_{j-1:k+1}(\mathcal{T}_{\mathbf{x}_{w_{j-1:k+1}}})N_{k,w_k}\|_H \\ &= \|\mathcal{T}_\rho^{1/2}\Pi_{t:j+1}(\mathcal{T}_\rho)\mathcal{T}_{\rho,\lambda}^{1/2}\mathcal{T}_{\rho,\lambda}^{-1/2}(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_{w_j}})\Pi_{j-1:k+1}(\mathcal{T}_{\mathbf{x}_{w_{j-1:k+1}}})\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\mathcal{T}_{\rho,\widetilde{\lambda}}^{-1/2}N_{k,w_k}\|_H \\ &\leq \|\mathcal{T}_\rho^{1/2}\Pi_{t:j+1}(\mathcal{T}_\rho)\mathcal{T}_{\rho,\lambda}^{1/2}\|\|\mathcal{T}_{\rho,\lambda}^{-1/2}(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_{w_j}})\|\|\Pi_{j-1:k+1}(\mathcal{T}_{\mathbf{x}_{w_{j-1:k+1}}})\|\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\|\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{-1/2}N_{k,w_k}\|_H \\ &\leq \|\mathcal{T}_\rho^{1/2}\Pi_{t:j+1}(\mathcal{T}_\rho)\mathcal{T}_{\rho,\lambda}^{1/2}\|\|\mathcal{T}_{\rho,\lambda}^{-1/2}(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_{w_j}})\|\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\|\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{-1/2}N_{k,w_k}\|_H \\ &\leq \|\mathcal{T}_\rho^{1/2}\Pi_{t:j+1}(\mathcal{T}_\rho)\mathcal{T}_{\rho,\lambda}^{1/2}\|\|\mathcal{T}_{\rho,\lambda}^{-1/2}(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_{w_j}})\|\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\|\max_{w\in V}\left\{\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{-1/2}N_{k,w}\|_H\right\},\end{aligned}$$

where we used, from $\eta\kappa^2 \leq 1$ and $\eta\|\mathcal{T}_{\mathbf{x}_v}\| \leq 1$ for any $v \in V$, that $\|\Pi_{j-1:k+1}(\mathcal{T}_{\mathbf{x}_{w_{j-1:k+1}}})\| \leq 1$ for $j \geq k+2$. The first operator norm is bounded as follows by using techniques similar to those used to prove Lemma 7:

$$\begin{aligned}\|\mathcal{T}_\rho^{1/2}\Pi_{t:j+1}(\mathcal{T}_\rho)\mathcal{T}_{\rho,\lambda}^{1/2}\| &\leq \left(\frac{1}{e\sum_{i=j+1}^t \eta_i} + \left(\frac{\lambda}{2e\sum_{i=j+1}^t \eta_i}\right)^{1/2}\right) \\ &\leq \left(\frac{1}{\sum_{i=j+1}^t \eta_i} + \left(\frac{\lambda}{\sum_{i=j+1}^t \eta_i}\right)^{1/2}\right).\end{aligned} \tag{15}$$

We proceed to construct a high-probability bound for the quantity $\|(\mathcal{T}_\rho + \lambda I)^{-1/2}(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_{w_j}})\|$, for any $w_j \in V$. For $v \in V$, let $\delta_v = \frac{\delta}{n}$ and apply (11) from Lemma 6 to obtain the following[5] with probability at least $1-\delta_v$:

$$\|(\mathcal{T}_\rho + \lambda I)^{-1/2}(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_v})\| \leq \|(\mathcal{T}_\rho + \lambda I)^{-1/2}(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_v})\|_{HS} \leq 2\kappa\left(\frac{2\kappa}{m\sqrt{\lambda}} + \frac{\sqrt{c_\gamma}}{\sqrt{m\lambda^\gamma}}\right)\log\frac{4n}{\delta}.$$

Applying a union bound yields the following with probability at least $1 - \sum_{v\in V}\delta_v = 1-\delta$:

$$\|(\mathcal{T}_\rho + \lambda I)^{-1/2}(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_v})\| \leq 2\kappa\left(\frac{2\kappa}{m\sqrt{\lambda}} + \frac{\sqrt{c_\gamma}}{\sqrt{m\lambda^\gamma}}\right)\log\frac{4n}{\delta} \quad \forall v \in V. \tag{16}$$

The result (13) then comes from plugging (15) and (16) into the expanded quantity at the start of the proof.

To prove (14), fix $t-1 \geq k \geq 1$ and $w_t, w_k \in V$. Expanding the norm we get

$$\begin{aligned}\|\mathcal{T}_\rho^{1/2}(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_{w_t}})N_{k,w_k}\|_H &= \|\mathcal{T}_\rho^{1/2}\mathcal{T}_{\rho,\lambda}^{1/2}\mathcal{T}_{\rho,\lambda}^{-1/2}(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_{w_t}})\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\mathcal{T}_{\rho,\widetilde{\lambda}}^{-1/2}N_{k,w_k}\|_H \\ &\leq \|\mathcal{T}_\rho^{1/2}\mathcal{T}_{\rho,\lambda}^{1/2}\|\|\mathcal{T}_{\rho,\lambda}^{-1/2}(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_{w_t}})\|\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\|\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{-1/2}N_{k,w_k}\|_H \\ &\leq \|\mathcal{T}_\rho^{1/2}\mathcal{T}_{\rho,\lambda}^{1/2}\|\|\mathcal{T}_{\rho,\lambda}^{-1/2}(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_{w_t}})\|\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\|\max_{w\in V}\left\{\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{-1/2}N_{k,w}\|_H\right\}.\end{aligned}$$

The result follows by using (16) to bound $\|\mathcal{T}_{\rho,\lambda}^{-1/2}(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_{w_t}})\|$. □

The following proposition utilise the previous proposition to bound the summation (12).

[5] For an operator L note that $\|L\| = \|LL^\star\|^{1/2}$ where $L^\star$ is the adjoint of L. The Hilbert-Schmidt norm bounds the operator norm as we have $\|L\|^2 = \|LL^\star\| \leq \mathrm{Tr}\left(LL^\star\right) = \|L\|_{HS}^2$.

Proposition 7. *Let the assumptions of Proposition 6 hold. For any $v \in V$, with probability at least $1-\delta$ we have*

$$\textbf{\textit{Resid. Emp. Cov. Error}} \leq 8\kappa\left(\frac{2\kappa}{m\sqrt{\lambda}} + \frac{\sqrt{c_\gamma}}{\sqrt{m\lambda^\gamma}}\right)\log\frac{4n}{\delta}\left[\boldsymbol{B}_1 + \boldsymbol{B}_2\right],$$

where

$$\boldsymbol{B}_1 = \|\mathcal{T}_\rho^{1/2}\mathcal{T}_{\rho,\lambda}^{1/2}\|\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\|\eta_t\sum_{k=1}^{t-1}\eta_k\max_{w\in V}\left\{\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{-1/2}N_{k,w}\|_H\right\},$$

$$\boldsymbol{B}_2 = \|\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\|\sum_{k=1}^{t-2}\eta_k\sum_{j=k+1}^{t-1}\eta_j\left(\frac{1}{\sum_{i=j+1}^t\eta_i} + \left(\frac{\lambda}{\sum_{i=j+1}^t\eta_i}\right)^{1/2}\right)\max_{w\in V}\left\{\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{-1/2}N_{k,w}\|_H\right\}.$$

Proof. Splitting the sum in (12) at $j = t$ and otherwise, directly applying (13) and (14) from Proposition 6 allows **Resid. Emp. Cov. Error** to be bounded as follows:

Resid. Emp. Cov. Error

$$\begin{aligned}
&\leq \eta_t\sum_{k=1}^{t-1}\eta_k\sum_{w_{t:k}\in V^{t-k+1}}|\Delta(w_{t:k})|\|\mathcal{T}_\rho^{1/2}(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}_{w_t}})\Pi_{t-1:k+1}(\mathcal{T}_{\mathbf{x}_{w_{t-1:k+1}}})N_{k,w_k}\|_H\\
&\quad+\sum_{k=1}^{t-2}\eta_k\sum_{w_{t:k}\in V^{t-k+1}}|\Delta(w_{t:k})|\sum_{j=k+1}^{t-1}\eta_j\|\mathcal{T}_\rho^{1/2}\Pi_{t:j+1}(\mathcal{T}_\rho)(\mathcal{T}_\rho-\mathcal{T}_{\mathbf{x}_{w_j}})\Pi_{j-1:k+1}(\mathcal{T}_{\mathbf{x}_{w_{j-1:k+1}}})N_{k,w_k}\|_H\\
&\leq 2\kappa\left(\frac{2\kappa}{m\sqrt{\lambda}} + \frac{\sqrt{c_\gamma}}{\sqrt{m\lambda^\gamma}}\right)\log\frac{4n}{\delta}\\
&\quad\times\Bigg[\underbrace{\|\mathcal{T}_\rho^{1/2}\mathcal{T}_{\rho,\lambda}^{1/2}\|\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\|\eta_t\sum_{k=1}^{t-1}\eta_k\max_{w\in V}\left\{\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{-1/2}N_{k,w}\|_H\right\}}_{\mathbf{B}_1}\sum_{w_{t:k}\in V^{t-k+1}}|\Delta(w_{t:k})|\\
&\qquad+\underbrace{\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\|\sum_{k=1}^{t-2}\eta_k\sum_{j=k+1}^{t-1}\eta_j\left(\frac{1}{\sum_{i=j+1}^t\eta_i} + \left(\frac{\lambda}{\sum_{i=j+1}^t\eta_i}\right)^{1/2}\right)\max_{w\in V}\left\{\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{-1/2}N_{k,w}\|_H\right\}}_{\mathbf{B}_2}\\
&\qquad\qquad\times\sum_{w_{t:k}\in V^{t-k+1}}|\Delta(w_{t:k})|\Bigg].
\end{aligned}$$

The result is then arrived at by applying the following bound for the summation $\sum_{w_{t:k}\in V^{t-k+1}}|\Delta(w_{t:k})|$ for each $k \leq t$:

$$\begin{aligned}
\sum_{w_{t:k}\in V^{t-k+1}}|\Delta(w_{t:k})| &= \sum_{w_{t:k}\in V^{t-k+1}}\left|P_{vw_{t:k}} - \frac{1}{n^{t-k+1}}\right|\\
&= \sum_{\substack{w_{t:k}\in V^{t-k+1}\\ P_{vw_{t:k}}\geq n^{-(t-k+1)}}}\left(P_{vw_{t:k}} - \frac{1}{n^{t-k+1}}\right) - \sum_{\substack{w_{t:k}\in V^{t-k+1}\\ P_{vw_{t:k}}< n^{-(t-k+1)}}}\left(P_{vw_{t:k}} - \frac{1}{n^{t-k+1}}\right) \leq 4.
\end{aligned}$$

□

Given Proposition 7 we can now plug in a high-probability bound for $\max_{w\in V}\left\{\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{-1/2}N_{k,w}\|_H\right\}$ and bound the resulting summations. This is summarised in the following lemma.

Lemma 9. *Let the assumptions of Proposition 6 hold with $0 \leq \theta \leq 3/4$, $0 \leq \lambda \leq \|\mathcal{T}_\rho\|$ and $0 \leq \widetilde{\lambda} \leq \|\mathcal{T}_\rho\|$. Given $\delta \in (0,1)$, the following holds with probability at least $1-\delta$:*

Resid. Emp. Cov. Error

$$\leq \widetilde{b}_1\frac{\log^2\frac{8n}{\delta}\log(t)}{m\sqrt{\left((m\lambda)\wedge\lambda^\gamma\right)\left((m\widetilde{\lambda})\wedge\widetilde{\lambda}^\gamma\right)}}(1\vee(\eta t^{1-\theta})\vee\sqrt{\lambda}(\eta t^{1-\theta})^{3/2}\vee(t^{-1}(\eta t^{1-\theta})^2)),$$

where $\widetilde{b}_1 = \frac{128\kappa(R\kappa^{2r}+\sqrt{M})(2\kappa+\sqrt{2\sqrt{\nu}c_\gamma})^2\|\mathcal{T}_\rho\|^{1/2}(4+\|\mathcal{T}_\rho\|)}{(1-\theta)}$.

Proof. Consider Proposition 7 with $\frac{\delta}{2}$, so the following holds with probability at least $1-\frac{\delta}{2}$

$$\begin{aligned}\textbf{Resid. Emp. Cov. Error} &\leq 8\kappa\left(\frac{2\kappa}{m\sqrt{\lambda}}+\frac{\sqrt{c_\gamma}}{\sqrt{m\lambda^\gamma}}\right)\log\frac{8n}{\delta}(\mathbf{B}_1+\mathbf{B}_2)\\ &\leq \frac{8\kappa(2\kappa+\sqrt{2\sqrt{\nu}c_\gamma})}{\sqrt{(m\lambda)\wedge\lambda^\gamma}}\frac{\log\frac{8n}{\delta}}{\sqrt{m}}(\mathbf{B}_1+\mathbf{B}_2),\end{aligned}$$

where we used that $\nu \geq 1$. Proceed to bound both $\mathbf{B}_1$ and $\mathbf{B}_2$. Start by constructing a high-probability bound for the term $\max_{w\in V}\left\{\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{-1/2}N_{k,w}\|_H\right\}\ k=1,\ldots,t$. For $v\in V$, let $\delta_v' = \frac{\delta}{2n}$. Lemma 6 states with probability at least $1-\delta_v'$ the following holds for any $k\in\mathbb{N}$:

$$\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{-1/2}N_{k,v}\| \leq 4(R\kappa^{2r}+\sqrt{M})\left(\frac{\kappa}{m\sqrt{\widetilde{\lambda}}}+\frac{\sqrt{2\sqrt{\nu}c_\gamma}}{\sqrt{m\widetilde{\lambda}^\gamma}}\right)\log\frac{8n}{\delta}.$$

Applying a union bound so the following holds with probability at least $1-\sum_{v\in V}\delta_v' = 1-\frac{\delta}{2}$ for any $k\in\mathbb{N}$:

$$\begin{aligned}\max_{w\in V}\left\{\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{-1/2}N_{k,w}\|_H\right\} &\leq 4(R\kappa^{2r}+\sqrt{M})\left(\frac{\kappa}{m\sqrt{\widetilde{\lambda}}}+\frac{\sqrt{2\sqrt{\nu}c_\gamma}}{\sqrt{m\widetilde{\lambda}^\gamma}}\right)\log\frac{8n}{\delta}\\ &\leq \frac{4(R\kappa^{2r}+\sqrt{M})(2\kappa+\sqrt{2\sqrt{\nu}c_\gamma})}{\sqrt{(m\widetilde{\lambda})\wedge\widetilde{\lambda}^\gamma}}\frac{\log\frac{8n}{\delta}}{\sqrt{m}},\end{aligned}\tag{17}$$

where we used that $\kappa \geq 1$. The terms $\mathbf{B}_1$ and $\mathbf{B}_2$ are now bounded in the following two paragraphs.

Term $\mathbf{B}_1$ Using the high-probability bound (17), the following holds with probability at least $1-\frac{\delta}{2}$:

$$\begin{aligned}\mathbf{B}_1 &\leq \|\mathcal{T}_\rho^{1/2}\mathcal{T}_{\rho,\lambda}^{1/2}\|\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\|\frac{4(R\kappa^{2r}+\sqrt{M})(2\kappa+\sqrt{2\sqrt{\nu}c_\gamma})}{\sqrt{(m\widetilde{\lambda})\wedge\widetilde{\lambda}^\gamma}}\frac{\log\frac{8n}{\delta}}{\sqrt{m}}\eta_t\sum_{k=1}^{t-1}\eta_k\\ &\leq \|\mathcal{T}_\rho^{1/2}\mathcal{T}_{\rho,\lambda}^{1/2}\|\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\|\frac{4(R\kappa^{2r}+\sqrt{M})(2\kappa+\sqrt{2\sqrt{\nu}c_\gamma})}{\sqrt{(m\widetilde{\lambda})\wedge\widetilde{\lambda}^\gamma}(1-\theta)}\frac{\log\frac{8n}{\delta}}{\sqrt{m}}t^{-1}(\eta t^{1-\theta})^2,\end{aligned}$$

where we have applied the integral bound t $\sum_{k=1}^{t-1}k^{-\theta}\leq\frac{t^{1-\theta}}{1-\theta}$, see for instance [27, Lemma 12], on the following summation:

$$\eta_t\sum_{k=1}^{t-1}\eta_k = \eta^2 t^{-\theta}\sum_{k=1}^{t-1}k^{-\theta} \leq \frac{\eta^2}{1-\theta}t^{1-2\theta} = \frac{t^{-1}(\eta t^{1-\theta})^2}{1-\theta}.$$

Term $\mathbf{B}_2$ Similarly, using the high-probability bound (17), the following holds with probability at least $1-\frac{\delta}{2}$:

$$\begin{aligned}\mathbf{B}_2 \leq \|\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\|&\frac{4(R\kappa^{2r}+\sqrt{M})(2\kappa+\sqrt{2\sqrt{\nu}c_\gamma})}{\sqrt{(m\widetilde{\lambda})\wedge\widetilde{\lambda}^\gamma}}\frac{\log\frac{8n}{\delta}}{\sqrt{m}}\\ &\times\sum_{k=1}^{t-2}\eta_k\sum_{j=k+1}^{t-1}\eta_j\left(\frac{1}{\sum_{i=j+1}^t\eta_i}+\left(\frac{\lambda}{\sum_{i=j+1}^t\eta_i}\right)^{1/2}\right).\end{aligned}$$

We proceed to bound the remaining terms by utilising results from Section C.5. Firstly, switching the order of sums and applying an integral bound yields

$$\begin{aligned}\sum_{k=1}^{t-2}\eta_k\sum_{j=k+1}^{t-1}\frac{\eta_j}{\sum_{i=j+1}^{t}\eta_i} &= \eta\sum_{k=1}^{t-2}k^{-\theta}\sum_{j=k+1}^{t-1}\frac{j^{-\theta}}{\sum_{i=j+1}^{t}i^{-\theta}}\\ &= \eta\sum_{j=2}^{t-1}\frac{j^{-\theta}}{\sum_{i=j+1}^{t}i^{-\theta}}\sum_{k=1}^{j-1}k^{-\theta}\\ &\le \frac{\eta}{1-\theta}\sum_{j=2}^{t-1}\frac{j^{-\theta}(j-1)^{1-\theta}}{\sum_{i=j+1}^{t}i^{-\theta}}. \end{aligned} \tag{18}$$

At this point use $\sum_{i=j+1}^{t} i^{-\theta} \ge t^{-\theta}(t-j)$ as well as Lemma 10 to obtain

$$\sum_{j=2}^{t-1}\frac{j^{-\theta}(j-1)^{1-\theta}}{\sum_{i=j+1}^{t}i^{-\theta}} \le t^{\theta}\sum_{j=2}^{t-2}\frac{(j-1)^{1-2\theta}}{t-j} \le 4t^{\theta}t^{-\min(2\theta-1,1)}\log(t) = 4t^{1-\theta}\log(t).$$

For the second term follow the steps to (18) and use Lemma 11 as follows:

$$\begin{aligned}\sum_{k=1}^{t-2}\eta_k\sum_{j=k+1}^{t-1}\frac{\eta_j}{\big(\sum_{i=j+1}^{t}\eta_i\big)^{1/2}} &\le \frac{\eta^{3/2}t^{\theta/2}}{1-\theta}\sum_{j=2}^{t-1}\frac{(j-1)^{1-2\theta}}{(t-j)^{1/2}}\\ &\le \frac{4\eta^{3/2}t^{\theta/2}}{1-\theta}t^{\max(3/2-2\theta,0)}\\ &= \frac{4\eta^{3/2}}{1-\theta}t^{\max(3(1-\theta)/2,\theta/2)}.\end{aligned}$$

This results in the following bound for $\mathbf{B}_2$, which holds with probability at least $1-\frac{\delta}{2}$:

$$\mathbf{B}_2 \le \|\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\|\frac{4(R\kappa^{2r}+\sqrt{M})(2\kappa+\sqrt{2\sqrt{\nu}c_\gamma})}{\sqrt{(m\widetilde{\lambda})\wedge\widetilde{\lambda}^{\gamma}}(1-\theta)}\frac{\log\frac{8n}{\delta}\log(t)}{\sqrt{m}}\big(4\eta t^{1-\theta}+4\sqrt{\lambda}\big(\eta t^{\max(1-\theta,\theta/3)}\big)^{3/2}\big).$$

The final bound arises by bringing everything together with a union bound implying it holds with probability at least $1-\frac{\delta}{2}-\frac{\delta}{2}=1-\delta$. Constants are then cleaned up using $\lambda \le \|\mathcal{T}_\rho\|$ as well as $\widetilde{\lambda} \le \|\mathcal{T}_\rho\|$ to say $\|\mathcal{T}_\rho^{1/2}\mathcal{T}_{\rho,\lambda}^{1/2}\|\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\| \le 4\|\mathcal{T}_\rho\|^{3/2}$ and $\|\mathcal{T}_{\rho,\widetilde{\lambda}}^{1/2}\| \le 2\|\mathcal{T}_\rho\|^{1/2}$. □

C.3.3 Network Error bound

In this section we bring together the bounds developed in the previous two sections for the **Population Covariance Error** term and **Residual Empirical Covariance Error** term to construct the final bound on the Network Term as presented in the following theorem.

Theorem 4. *Let Assumptions 1, 2, 3 hold with $r \ge 1/2$, and $\eta_t = \eta t^{-\theta}$ for $t \in \mathbb{N}$ with $\eta\kappa^2 \le 1$ and $\theta \in (0, 3/4)$. Assume $t/2 \ge \lceil\frac{(r+1)\log(t)}{1-\sigma_2}\rceil =: t^\star$ The following bound holds for any $v \in V$, $\alpha \in [0, 1/2]$ and $\gamma' \in [1, \gamma]$:*

$$\begin{aligned}&\mathbf{E}[\|\mathcal{S}_\rho(\omega_{t+1,v}-\xi_{t+1,v})\|_\rho^2] \le 2\frac{\widetilde{a}\log^2(4n)\log^2(t^\star)}{m}\Big(\eta^2t^{-2r}\vee(m^{-1}(\eta t^\star)^{1+2\alpha})\vee(\eta t^\star)^{\gamma'+2\alpha}\Big)\\ &+2\widetilde{b}_2\frac{\log^4(8n)\log^2(t)}{m^2}\Big(1\vee(\eta t^{1-\theta})^2\vee(t^{-2}(\eta t^{1-\theta})^4)\Big)\Big((m^{-1}\eta t^{1-\theta})\vee(\eta t^{1-\theta})^\gamma\Big),\end{aligned}$$

where $\widetilde{b}_2 = 64\frac{(\|\mathcal{T}_\rho\|+1)^2}{(\|\mathcal{T}_\rho\|\wedge\|\mathcal{T}_\rho\|^\gamma)^2}\widetilde{b}_1^2$ with $\widetilde{b}_1$ defined as in Theorem 9 and $\widetilde{a}$ defined as in Lemma 8.

Proof. Use decomposition (9). Taking the expectation, note that the first term $\mathbf{E}[(\textbf{Pop. Cov. Error})^2]$ is controlled by Lemma 8. We now proceed to control the term $\mathbf{E}[(\textbf{Resid. Emp. Cov. Error})^2]$.

Begin by using the high-probability bound for **Resid. Emp. Cov. Error** in Lemma 9, with $\widetilde{\lambda} = \|\mathcal{T}_\rho\|$ and $\lambda = \|\mathcal{T}_\rho\|(\eta t^{1-\theta})^{-1}$. The following upper bound holds for the quantity that appears in Lemma 9:

$$\begin{aligned}\frac{1}{(m\lambda)\wedge\lambda^\gamma} &= \frac{1}{(\|\mathcal{T}_\rho\| m(\eta t^{1-\theta})^{-1})\wedge(\|\mathcal{T}_\rho\|^\gamma(\eta t^{1-\theta})^{-\gamma})} \\ &\leq \frac{1}{\|\mathcal{T}_\rho\|\wedge\|\mathcal{T}_\rho\|^\gamma}\Big((m^{-1}(\eta t^{1-\theta}))\vee(\eta t^{1-\theta})^\gamma\Big).\end{aligned}$$

Plugging the above into Lemma 9 for the **Resid. Emp. Cov. Error** allows the expectation to be bounded with Lemma 4:

$$\begin{aligned}&\mathbf{E}[(\textbf{Resid. Emp. Cov. Error})^2] \\ &\leq \widetilde{b}_1^2\frac{(\|\mathcal{T}_\rho\|+1)^2}{\big(\|\mathcal{T}_\rho\|\wedge\|\mathcal{T}_\rho\|^\gamma\big)^2}\frac{\log^2(t)}{m^2}\Big(1\vee(\eta t^{1-\theta})^2\vee(t^{-2}(\eta t^{1-\theta})^4)\Big)\Big((m^{-1}\eta t^{1-\theta})\vee(\eta t^{1-\theta})^\gamma\Big) \\ &\quad\times\int_0^1\log^4\frac{8n}{\delta}d\delta.\end{aligned}$$

The result is arrived at by using $\int_0^1\log^4\frac{8n}{\delta}d\delta \leq 64\log^4(8n)$ and bringing together the two bounds for $\mathbf{E}[(\textbf{Pop. Cov. Error})^2]$ and $\mathbf{E}[(\textbf{Resid. Emp. Cov. Error})^2]$. □

C.4 Final Bound

In this section we bring together the bounds from the previous sections to construct the final bounds in Theorem 2 and Theorem 1 in the main body of the work. The main result is the following.

Theorem 5. *Let Assumptions 1, 2, 3 hold with $r \geq 1/2$ and $\eta_t = \eta t^{-\theta}$ for all $t \in \mathbb{N}$ with $\eta\kappa^2 \leq 1$ $\theta \in (0, 3/4)$. The following holds for all $t/2 \geq \lceil\frac{(r+1)\log(t)}{1-\sigma_2}\rceil =: t^\star$, any $v \in V$, $\alpha \in [0, 1/2]$ and $\gamma' \in [1, \gamma]$:*

$$\begin{aligned}&\mathbf{E}[\mathcal{E}(\omega_{t+1,v})] - \inf_{\omega\in H}\mathcal{E}(\omega) \leq 2R^2(\eta t^{1-\theta})^{-2r} \\ &+ \widetilde{d}_4(nm)^{-2r/(2r+\gamma)}\Big(1\vee(nm)^{-2/(2r+\gamma)}(\eta t^{1-\theta})^2\vee t^{-2}(\eta t^{1-\theta})^2\Big)\log^2(t) \\ &+ 8\frac{\widetilde{a}\log^2(4n)\log^2(t^\star)}{m}\Big(\eta^2 t^{-2r}\vee(m^{-1}(\eta t^\star)^{1+2\alpha})\vee(\eta t^\star)^{\gamma'+2\alpha})\Big) \\ &+ 8\frac{\widetilde{b}_2\log^4(8n)\log^2(t)}{m^2}\Big(1\vee(\eta t^{1-\theta})^2\vee t^{-2}(\eta t^{1-\theta})^4\Big)\Big((m^{-1}\eta t^{1-\theta})\vee(\eta t^{1-\theta})^\gamma\Big),\end{aligned}$$

where $\widetilde{d}_4 = 4\big(\frac{2r+\gamma}{2r+\gamma-1}\big)^2\widetilde{d}_3^2$ with $\widetilde{d}_3$ defined as in Theorem 3.

Proof. Begin with the decomposition in Proposition 1 and take the expectation $\mathbf{E}[\,\cdot\,]$. Plug in the bounds for each term proven in the previous sections, i.e. Proposition 2 for the Bias, Theorem 3 with $p = 1/(2r+\gamma)$ for the Sample Variance term and Theorem 4 for the Network Error term. □

Theorem 2 follows directly from Theorem 5.

Proof of Theorem 2. Consider Theorem 5 with constants

$$\begin{aligned}q_1 &= 2R^2 \\ q_2 &= \widetilde{d}_4 \\ q_3 &= 16\widetilde{a}(\log^2(4)+1) \\ q_4 &= 24\widetilde{b}_2(\log^2(8)+1)^2,\end{aligned}$$

where the sample variance constant $\widetilde{d}_4$ is defined in Theorem 5, the first network error constant $\widetilde{a}$ is defined in Lemma 8, and the second network error constant $\widetilde{b}_2$ is defined in Theorem 4. □

We now go on to prove Theorem 1.

Proof of Theorem 1. Consider the setting of Theorem 5 with $\theta = 0$. Begin by setting

$$t = \Big\lceil (nm)^{1/(2r+\gamma)} \Big[\frac{1}{1-\sigma_2} \Big(\frac{n^r}{m^{r+\gamma}} \Big)^{2/((1+2\alpha)(2r+\gamma))} \vee \frac{1}{1-\sigma_2} \Big(\frac{n^{2r}}{m^{\gamma}} \Big)^{1/((\gamma'+2\alpha)(2r+\gamma))} \vee 1 \Big] \Big\rceil$$

and $\eta = \kappa^{-2}(nm)^{1/(2r+\gamma)}/t$. It is clear that $\eta t = \kappa^{-2}(nm)^{1/(2r+\gamma)}$. We proceed to show that this choice of iterations t and step size η ensures each of the terms in the bound of Theorem 5 are of order $\widetilde{O}((nm)^{-2r/(2r+\gamma)})$.

The Bias term is

$$2R^2(\eta t)^{-2r} = 2R^2\kappa^{4r}(nm)^{-2r/(2r+\gamma)}.$$

The Sample Variance term is bounded as follows:

$$\begin{aligned}
&\widetilde{d}_4(nm)^{-2r/(2r+\gamma)} \Big(1 \vee (nm)^{-2/(2r+\gamma)}(\eta t)^2 \vee t^{-2}(\eta t)^2 \Big) \log^2(t) \\
&\leq 4\kappa^{-4}\widetilde{d}_4(nm)^{-2r/(2r+\gamma)} \log^2(t).
\end{aligned}$$

The first Network Error term is bounded in three parts aligning with the three terms within the quantity $m^{-1}(\eta^2 t^{-2r} \vee (m^{-1}(\eta t^\star)^{1+2\alpha}) \vee (\eta t^\star)^{\gamma'+2\alpha}))$. Firstly, as $t \geq (nm)^{1/(2r+\gamma)}$ and $\eta \leq 1/\kappa^2$ we get $\eta^2 t^{-2r} \leq \kappa^{-4}(nm)^{-2r/(2r+\gamma)}$. Secondly, from $t \geq (nm)^{1/(2r+\gamma)} \frac{1}{1-\sigma_2} \Big(\frac{n^r}{m^{r+\gamma}} \Big)^{2/((1+2\alpha)(2r+\gamma))}$ ensuring $\eta \leq \kappa^{-2}(1-\sigma_2) \Big(\frac{m^{r+\gamma}}{n^r} \Big)^{2/((1+2\alpha)(2r+\gamma))}$ we get

$$\begin{aligned}
\frac{(\eta t^\star)^{1+2\alpha}}{m^2} &\leq (\kappa^{-2} 2(r+1)\log(t))^{1+2\alpha} \frac{m^{2(r+\gamma)/(2r+\gamma)-2}}{n^{2r/(2r+\gamma)}} \\
&= (\kappa^{-2} 2(r+1)\log(t))^{1+2\alpha} (nm)^{-2r/(2r+\gamma)}.
\end{aligned}$$

Thirdly, from $t \geq (nm)^{1/(2r+\gamma)} \frac{1}{1-\sigma_2} \Big(\frac{n^{2r}}{m^{\gamma}} \Big)^{1/((\gamma'+2\alpha)(2r+\gamma))}$ we have $\eta \leq \kappa^{-2}(1-\sigma_2) \Big(\frac{m^{\gamma}}{n^{2r}} \Big)^{1/((\gamma'+2\alpha)(2r+\gamma))}$ and so

$$\begin{aligned}
\frac{(\eta t^\star)^{\gamma'+2\alpha}}{m} &\leq (\kappa^{-2} 2(r+1)\log(t))^{\gamma'+2\alpha} \frac{m^{\gamma/(2r+\gamma)-1}}{n^{2r/(2r+\gamma)}} \\
&= (\kappa^{-2} 2(r+1)\log(t))^{\gamma'+2\alpha} (nm)^{-2r/(2r+\gamma)}.
\end{aligned}$$

Using the above three bounds we arrive at the first Network term being $\widetilde{O}((nm)^{-2r/(2r+\gamma)})$.

Now consider the second Network Error term. Since $\eta t = \kappa^{-2}(nm)^{1/(2r+\gamma)}$ and $m \geq n^{\frac{2r+2+\gamma}{2r+\gamma-2}} \geq n^{\frac{1-\gamma}{2(r+\gamma)-1}}$ we have

$$\Big(1 \vee (\eta t)^2 \vee t^{-2}(\eta t)^4 \Big) \Big(\big(m^{-1}(\eta t) \big) \vee (\eta t)^\gamma \Big) \leq \Big(1 \vee (\eta t)^{2+\gamma} \vee t^{-2}(\eta t)^{4+\gamma} \Big).$$

The second Network Error term then becomes, due to $t \geq (nm)^{1/(2r+\gamma)}$,

$$\begin{aligned}
&8\frac{\widetilde{b}_2 \log^4(8n)\log^2(t)}{m^2} \Big(1 \vee (\eta t)^{2+\gamma} \vee t^{-2}(\eta t)^{4+\gamma} \Big) \\
&\leq 8(\kappa^{-2})^{2+\gamma} \widetilde{b}_2 \log^4(8n)\log^2(t) \frac{(nm)^{(2+\gamma)/(2r+\gamma)}}{m^2}.
\end{aligned}$$

For this quantity to be $\widetilde{O}((nm)^{-2r/(2r+\gamma)})$ we require $\frac{(nm)^{(2+\gamma)/(2r+\gamma)}}{m^2} \leq (nm)^{-2r/(2r+\gamma)}$ which is satisfied for $m \geq n^{(2r+\gamma+2)/(2r+\gamma-2)}$. Now ensure $\frac{t}{\log(t)} \geq 2\frac{(1+r)}{1-\sigma_2}$. Note the previous requirements on the iterations t imply

$$t \geq \frac{(nm)^{1/(2r+\gamma)}}{1-\sigma_2} \frac{n^{2r/(2r+\gamma)}}{m^{\gamma/(2r+\gamma)}} \geq \frac{n^{(2r+1)/2r+\gamma}}{1-\sigma_2} \geq \frac{n}{1-\sigma_2}.$$

And since $x \to x/(\log(x))$ is increasing for $x \geq 1$, the requirement $t \geq 2\frac{(1+r)\log(t)}{(1-\sigma_2)}$ is satisfied by $\frac{n}{\log(\frac{n}{1-\sigma_2})} \geq 2(1+r)$.

Now, consider choosing $\gamma' \in [1, \gamma]$ and $\alpha \in [0, 1/2]$ to minimise the number of iterations t. Consider the two cases $m \geq n^{2r/\gamma}$ and $m \leq n^{2r/\gamma}$. When $m \geq n^{2r/\gamma}$ we have both $\frac{n^{2r}}{m^\gamma} \leq 1$ and $\frac{n^r}{m^{r+\gamma}} \leq 1$ so the number of iterations t required is minimised by picking $\gamma' = \gamma$ and $\alpha = 0$. Since $2(r+\gamma) \geq 1$ we get $\frac{n^{2r}}{m^{2(r+\gamma)}} \leq \frac{n^{2r/\gamma}}{m}$ and the number of iterations becomes $t = (nm)^{1/(2r+\gamma)}\left[\left(\frac{1}{1-\sigma_2}\left(\frac{n^{2r/\gamma}}{m}\right)^{1/(2r+\gamma)}\right) \vee 1\right] = (nm)^{1/(2r+\gamma)}\left[\left(\frac{(nm)^{2r/(2r+\gamma)}}{m(1-\sigma_2)^\gamma}\right)^{1/\gamma} \vee 1\right]$. When $\frac{n^{2r}}{m^\gamma} \geq 1$, the number of iterations t required is minimised by: setting $\gamma' = 1$, noting $\frac{n^{2r}}{m^{2(r+\gamma)}} \leq \frac{n^{2r}}{m^\gamma}$ and further picking $\alpha = 1/2$. It is clear in this case that the number of iterations required becomes $t = (nm)^{1/2r+\gamma}\frac{1}{1-\sigma_2}\left(\frac{n^r}{m^{\gamma/2}}\right)^{1/(2r+\gamma)} = (nm)^{1/(2r+\gamma)}\frac{(nm)^{r/(2r+\gamma)}}{\sqrt{m}(1-\sigma_2)}$.

□

C.5 Useful inequalities

In this section we collect useful inequalities used within the proofs.

Lemma 10. *The following holds for $q \in \mathbb{R}$ and $t \in \mathbb{N}$ with $t \geq 3$:*

$$\sum_{k-1}^{t-1} \frac{1}{t-k} k^{-q} \leq 2t^{-\min(q,1)}(1 + \log(t)).$$

Proof. See Lemma 14 in [27]. □

Lemma 11. *The following holds for $q \in \mathbb{R}$ and $t \in \mathbb{N}$ with $t \geq 3$:*

$$\sum_{k-1}^{t-1} \frac{1}{(t-k)^{1/2}} k^{-q} \leq 4t^{\max(1/2-q,0)}.$$

Proof. Begin with

$$\sum_{k=1}^{t-1} \frac{1}{(t-k)^{1/2}} k^{-q} \leq t^{\max(1/2-q,0)} \sum_{k=1}^{t-1} \frac{1}{(t-k)^{1/2}k^{1/2}}.$$

Suppose t is even. The bound arises by splitting the sum and using the integral bounds

$$\sum_{k=1}^{t/2} \frac{1}{(t-k)^{1/2}k^{1/2}} \leq \frac{\sqrt{2}}{t^{1/2}} \sum_{k=1}^{t/2} \frac{1}{k^{1/2}} \leq \frac{\sqrt{2}}{t^{1/2}}\left[1 + \int_1^{t/2} x^{-1/2} dx\right] = \frac{\sqrt{2}}{t^{1/2}}\left[1 + 2\left(\sqrt{\frac{t}{2}} - 1\right)\right] \leq 2,$$

and

$$\begin{aligned}\sum_{k=t/2+1}^{t-1} \frac{1}{(t-k)^{1/2}k^{1/2}} &\leq \sqrt{\frac{2}{t}} \sum_{k=t/2+1}^{t-1} \frac{1}{(t-k)^{1/2}} \leq \sqrt{\frac{2}{t}}\left[1 + \int_{t/2+1}^{t-1} (t-x)^{-1/2} dx\right] \\ &= \sqrt{\frac{2}{t}}\left[1 + 2\left(\sqrt{\frac{t}{2} - 1} - 1\right)\right] \leq 2.\end{aligned}$$

If t is odd, follow the steps above and split the sum at $k = (t-1)/2$ and $k = (t-1)/2 + 1$. □

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Optimal Statistical Rates for Decentralised Non-Parametric Regression with Linear Speed-Up
Publication Status	☐Published
Publication Details	Optimal Statistical Rates for Decentralised Non-ParametricRegression with Linear Speed-Up", Dominic Richards, Patrick Rebeschini. In Advances in Neural Information Processing Systems, 2019.

Student Confirmation

Student Name:	Dominic Richards		
Contribution to the Paper	Derived the main result, which is a bound on test error for distributed gradient descent. Wrote first draft of manuscript and assisted writing final version alongside Patrick Rebeschini.		
Signature	Dominic Richards	Date	04/01/2021

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	PATRICK REBESCHINI ASSOCIATE PROFESSOR		
Supervisor comments			
Signature	Patrick Rebeschini	Date	11/01/2021

This completed form should be included in the thesis, at the end of the relevant chapter.

4

Decentralised Learning with Random Features and Distributed Gradient Descent

Decentralised Learning with Distributed Gradient Descent and Random Features

Dominic Richards [1] **Patrick Rebeschini** [1] **Lorenzo Rosasco** [2 3 4]

Abstract

We investigate the generalisation performance of Distributed Gradient Descent with Implicit Regularisation and Random Features in the homogenous setting where a network of agents are given data sampled independently from the same unknown distribution. Along with reducing the memory footprint, Random Features are particularly convenient in this setting as they provide a common parameterisation across agents that allows to overcome previous difficulties in implementing Decentralised Kernel Regression. Under standard source and capacity assumptions, we establish high probability bounds on the predictive performance for each agent as a function of the step size, number of iterations, inverse spectral gap of the communication matrix and number of Random Features. By tuning these parameters, we obtain statistical rates that are minimax optimal with respect to the total number of samples in the network. The algorithm provides a linear improvement over single machine Gradient Descent in memory cost and, when agents hold enough data with respect to the network size and inverse spectral gap, a linear speed-up in computational runtime for any network topology. We present simulations that show how the number of Random Features, iterations and samples impact predictive performance.

1. Introduction

In supervised learning, an agent is given a collection of training data to fit a model that can predict the outcome of new data points. Due to the growing size of modern data sets and complexity of many machine learning models, a popular approach is to incrementally improve the model with respect to a loss function that measures the performance on the training data. The complexity and stability of the resulting model is then controlled implicitly by algorithmic parameters, such as, in the case of Gradient Descent, the step size and number of iterations. An appealing collection of models in this case are those associated to the Reproducing Kernel Hilbert Space (RKHS) for some positive definite kernel, as the resulting optimisation problem (originally over the space of functions) admits a tractable form through the Kernel Trick and Representer Theorem, see for instance (Schölkopf et al., 2001).

Given the growing size of data, privacy concerns as well as the manner in which data is collected, distributed computation has become a requirement in many machine learning applications. Here training data is split across a number of agents which alternate between communicating model parameters to one another and performing computations on their local data. In centralised approaches (effective star topology), a single agent is typically responsible for collecting, processing and disseminating information to the agents. Meanwhile for many applications, including ad-hoc wireless and peer-to-peer networks, such centralised approaches are unfeasible. This motivates decentralised approaches where agents in a network only communicate locally within the network i.e. to neighbours at each iteration.

Many problems in decentralised multi-agent optimisation can be phrased as a form of consensus optimisation (Tsitsiklis et al., 1986; Tsitsiklis, 1984; Johansson et al., 2007; Nedic & Ozdaglar, 2009; Nedić et al., 2009; Johansson et al., 2009; Lobel & Ozdaglar, 2011; Matei & Baras, 2011; Boyd et al., 2011; Duchi et al., 2012; Shi et al., 2015; Mokhtari & Ribeiro, 2016). In this setting, a network of agents wish to minimise the average of functions held by individual agents, hence "reaching consensus" on the solution of the global problem. A standard approach is to augment the original optimisation problem to facilitate a decentralised algorithm. This typically introduces additional penalisation (or constraints) on the difference between neighbouring agents within the network, and yields a higher dimensional optimisation problem which decouples across the agents.

[1]Department of Statistics, University of Oxford, 24-29 St Giles', Oxford, OX1 3LB [2]MaLGa Center, Universitá degli Studi di Genova, Genova, Italy [3]Istituto Italiano di Tecnologia, Via Morego, 30, Genoa 16163, Italy [4]Massachusetts Institute of Technology, Cambridge, MA 02139, USA. Correspondence to: Patrick Rebeschini <patrick.rebeschini@stats.ox.ac.uk>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

This augmented problem can then often be solved using standard techniques whose updates can now be performed in a decentralised manner. While this approach is flexible and can be applied to many consensus optimisation problems, it often requires more complex algorithms which depend upon the tuning of additional hyper parameters, see for instance the Alternating Direction Method of Multiplers (ADMM) (Boyd et al., 2011).

Many distributed machine learning problems, in particular those involving empirical risk minimisation, can been framed in the context of consensus optimisation. As discussed in (Bouboulis et al., 2017; Koppel et al., 2018), for the case of Decentralised Kernel Regression it is not immediately clear how the objective ought to be augmented to facilitate both a decentralised algorithm and the Representer Theorem. Specifically, so the problem decouples across the network and agents have a common represention of the estimated function. Indeed, while distributed kernel regression can be performed in the one-shot Divide and Conquer setting (Star Topology) (Zhang et al., 2015; Lin et al., 2017; Guo et al., 2017; Mücke & Blanchard, 2018; Dobriban & Sheng, 2020) where there is a fusion center to combine the resulting estimators computed by each agent, in the decentralised setting there is no fusion center and agents must communicate for multiple rounds. A number of works have aimed to tackle this challenge (Forero et al., 2010; Mitra & Bhatia, 2014; Gao et al., 2015; Chouvardas & Draief, 2016; Bouboulis et al., 2017; Koppel et al., 2018), although these methods often include approximations whose impact on statistical performance is not clear[1]. Most relevant to our work is (Bouboulis et al., 2017) where Distributed Gradient Descent with Random Fourier Features is investigated in the online setting. In this case regret bounds are proven, but it is not clear how the number of Random Fourier Features or network topology impacts predictive performance in conjunction with non-parametric statistical assumptions[2]. For more details on the challenges of the developing a Decentralised Kernel Regression algorithm see Section 2.1.

1.1. Contributions

In this work we give statistical guarantees for a simple and practical Decentralised Kernel Regression algorithm. Specifically, we study the learning performance (Generalisation Error) of full-batch Distributed Gradient Descent (Nedic & Ozdaglar, 2009) with implicit regularisation (Richards & Patrick, 2020; Richards & Rebeschini, 2019) and Random Features (Rahimi & Recht, 2008; Rudi & Rosasco, 2017). Random Features can be viewed as a form of non-linear sketching or shallow neural networks with random initialisations, and have be utilised to facilitate the large scale application of kernel methods by overcoming the memory bottle-neck. In our case, they both decrease the memory cost and yield a simple Decentralised Kernel Regression algorithm. While previous approaches have viewed Decentralised Kernel Regression with explicit regularisation as an instance of consensus optimisation, where the speed-up in runtime depends on the network topology (Duchi et al., 2012; Scaman et al., 2017). We build upon (Richards & Rebeschini, 2019) and directly study the Generalisation Error of Distributed Gradient Descent with implicit regularisation. This allows linear speed-ups in runtime for *any* network topology to be achieved by leveraging the statistical concentration of quantities held by agents. Specifically, our analysis demonstrates how the number of Random Features, network topology, step size and number of iterations impact Generalisation Error, and thus, can be tuned to achieve minimax optimal statistical rates with respect to all of the samples within the network (Caponnetto & De Vito, 2007). When agents have sufficiently many samples with respect to the network size and topology, and the number of Random Features equal the number required by single machine Gradient Descent, a linear speed-up in runtime and linear decrease memory useage is achieved over single machine Gradient Descent. Previous guarantees given in consensus optimisation require the number of iterations to scale with the inverse spectral gap of the network (Duchi et al., 2012; Scaman et al., 2017), and thus, a linear speed-up in runtime is limited to well connected topologies. We now provide a summary of our contributions.

- **Decentralised Kernel Regression Algorithm**: By leveraging Random Features we develop a simple, practical and theoretically justified algorithm for Decentralised Kernel Regression. It achieves a linear reduction in memory cost and, given sufficiently many samples, a linear speed-up in runtime for any graph topology (Theorem 1, 2). This required extending the theory of Random Features to the decentralised setting (Section 4).
- **Refined Statistical Assumptions**: Considering the attainable case in which the minimum error over the hypothesis class is achieved, we give guarantees that hold over a wider range of complexity and capacity assumptions. This is achieved through a refined analysis of the Residual Network Error term (Section 4.4).
- **Bounds in High Probability**: All guarantees hold in high probability, where previous results (Richards & Rebeschini, 2019) for the decentralised setting only held in expectation. This is achieved through refined analysis of the Population Network Error (Section 4.3).

[1] Additional details on some of these works have been included within Remark 2 in the Appendix

[2] We note the concurrent work (Xu et al., 2020) which also investigates Random Fourier Features for decentralised non-parametric learning. The differences from our work have been highlighted in Remark 3 in the Appendix.

This work is structured as follows. Section 2 introduces the notation and Random Features. Section 3 presents the main theoretical results. Section 4 provides the error decomposition and a sketch proof of the refined analysis. Section 5 presents simulation results. Section 6 gives the conclusion.

2. Setup

This section introduces the setting. Section 2.1 introduces Decentralised Kernel Regression and the challenges in developing a decentralised algorithm. Section 2.2 introduces the link between Random Features and kernel methods. Section 2.3 introduces Distributed Gradient Descent with Random Features.

2.1. Challenges of Decentralised Kernel Regression

We begin with the single machine case then go on to the decentralised case.

Single Machine Consider a standard supervised learning problem with squared loss. Given a probability distribution ρ over $X \times \mathbb{R}$, we wish to solve

$$\min_f \mathcal{E}(f), \quad \mathcal{E}(f) = \int (f(x) - y)^2 d\rho(x, y), \qquad (1)$$

given a collection of independently and identically distributed (i.i.d.) samples drawn from ρ, here denoted $(x_i, y_i)_{i=1}^m \in (X \times \mathbb{R}^m)$. Kernel methods are non-parametric approaches defined by a kernel $k : X \times X \to \mathbb{R}$ which is symmetric and positive definite. The space of functions considered will be the Reproducing Kernel Hilbert Space associated to the kernel k, that is, the function space $\mathcal{H}$ defined as the completion of the linear span $\{K(x, \cdot) : x \in X\}$ with respect to the inner product $\langle K(x, \cdot), K(x', \cdot)\rangle_{\mathcal{H}} := K(x, x')$ (Aronszajn, 1950). When considering functions that minimise the empirical loss with explicit regularisation $\lambda \geq 0$

$$\min_{f \in \mathcal{H}} \left\{ \frac{1}{m} \sum_{i=1}^m (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2 \right\} \qquad (2)$$

we can appeal to the Representer Theorem (Schölkopf et al., 2001), and consider functions represented in terms of the data points, namely $\widehat{f}(x) = \sum_{i=1}^m \alpha_i k(x_i, x)$ where $\alpha = (\alpha_1, \ldots, \alpha_m) \in \mathbb{R}^m$ are a collection of weights. The weights are then often written in terms of the gram-matrix $K \in \mathbb{R}^{m \times m}$ whose i, jth entry is $K_{ij} = k(x_i, x_j)$.

Decentralised Consider a connected network of n agents $G = (V, E)$ $|V| = n$, joined by edges $E \subseteq V \times V$, that wish to solve (1). Each agent $v \in V$ has a collection of m i.i.d. training points $(x_{i,v}, y_{i,v})_{i=1}^m \in (X \times \mathbb{R})^m$ sampled from ρ. Following standard approaches in consensus optimisation we arrive at the optimisation problem

$$\min_{f_v \in \mathcal{H}, v \in V} \left\{ \frac{1}{nm} \sum_{v \in V} \sum_{i=1}^m (f_v(x_{i,v}) - y_{i,v})^2 + \lambda \|f_v\|_{\mathcal{H}}^2 \right\}$$
$$f_v = f_w \quad (v, w) \in E,$$

where a local function for each agent f_v is only evaluated at the data held by that agent $(x_{i,v}, y_{i,v})_{i=1}^m$, and a constraint ensures agents that share an edge are equal. This constrained problem is then often solved by considering the dual problem (Scaman et al., 2017) or introducing penalisation (Jakovetić et al., 2015). In either case, the objective decouples so that given $\{f_v\}_{v \in V}$ it can be evaluated and optimised in a decentralised manner. As discussed by (Bouboulis et al., 2017; Koppel et al., 2018), it is not immediately clear whether a representation for $\{f_v\}_{v \in V}$ exists in this case that respects the gram-matrices held by each agent. Recall, in the decentralised setting, only agent v can access the data $(x_{i,v}, y_{i,v})_{i=1}^m$ and the kernel evaluated at their data points $k(x_{i,v}, x_{j,v})$ for $i, j = 1, \ldots, m$.

2.2. Feature Maps and Kernel Methods

Consider functions parameterised by $\omega \in \mathbb{R}^M$ and written in the following form

$$f(x) = \langle \omega, \phi_M(x) \rangle, \quad \forall x \in X,$$

where $\phi_M : X \to \mathbb{R}^M, M \in \mathbb{N}$, denotes a family of finite dimensional feature maps that are identical and known across all of the agents. Feature maps in our case take a data point x to a (often higher dimensional) space where Euclidean inner products approximate the kernel. That is, informally, $k(x, x') \approx \langle \phi_M(x), \phi_M(x') \rangle$. One now classical example is Random Fourier Features (Rahimi & Recht, 2008) which approximate the Gaussian Kernel.

Random Fourier Features If $k(x, x') = G(x - x')$, where $G(z) = e^{-\frac{1}{2\sigma^2}\|z\|^2}$, for $\sigma > 0$ then we have

$$G(x - x') = \frac{1}{2\pi Z} \int \int_0^{2\pi}$$
$$\sqrt{2}\cos(\omega^\top x + b)\sqrt{2}\cos(\omega^\top x' + b) e^{-\frac{\sigma^2}{2}\|\omega\|^2} d\omega db$$

where Z is a normalizing factor. Then, for the Gaussian kernel, $\phi_M(x) = M^{-1/2}(\sqrt{2}\cos(\omega_1^\top x + b_1), \ldots, \sqrt{2}\cos(\omega_M^\top x + b_M))$, where $\omega_1, \ldots, \omega_M$ and $b_1, \ldots, b_M$ sampled independently from $\frac{1}{Z} e^{-\sigma^2\|\omega\|^2/2}$ and uniformly in $[0, 2\pi]$, respectively.

More generally, this motivates the strategy in which we assume the kernel k can be expressed as

$$k(x, x') = \int \psi(x, \omega)\psi(x', \omega) d\pi(\omega), \quad \forall x, x' \in X, \quad (3)$$

where (Ω, π) is a probability space and $\psi : X \times \Omega \to \mathbb{R}$ (Reed, 2012). Random Features can then be seen as Monte Carlo approximations of the above integral.

2.3. Distributed Gradient Descent and Random Features

Since the functions are now linearly parameterised by $\omega \in \mathbb{R}^M$, agents can consider the simple primal method Distributed Gradient Descent (Nedic & Ozdaglar, 2009). Initialised at $\widehat{\omega}_{1,v} = 0$; for $v \in V$, agents update their iterates for $t \geq 1$

$$\widehat{\omega}_{t+1,v} = \sum_{w \in V} P_{vw} \times \Big(\widehat{\omega}_{t,w} - \frac{\eta}{m} \sum_{i=1}^{m} \big(\langle \widehat{\omega}_{t,w}, \phi_M(x_{i,w}) \rangle - y_{i,w} \big) \phi_M(x_{i,w}) \Big), \tag{4}$$

where $P \in \mathbb{R}^{n \times n}$ is a doubly stochastic matrix supported on the network i.e. $P_{ij} \neq 0$ only if $(i,j) \in E$, and η is a fixed stepsize. The above iterates are a combination of two steps. Each agent performing a local Gradient Descent step with respect to their own data i.e. $\widehat{\omega}_{t,w} - \frac{\eta}{m} \sum_{i=1}^{m} \big(\langle \widehat{\omega}_{t,w}, \phi_M(x_{i,w}) \rangle - y_{i,w} \big) \phi_M(x_{i,w})$ for agent $w \in V$. And a communication step where agents average with their neighbours as encoded by the summation $\sum_{w \in V} P_{vw} a_w$, where a_w is the quantity held by agent $w \in V$. The performance of Distributed Gradient Descent naturally depends on the connectivity of the network. In our case it is encoded by the second largest eigenvalue of P in absolute value, denoted $\sigma_2 \in [0, 1)$. In particular, it arises through the inverse spectral gap $1/(1-\sigma_2)$, which is known to scale with the network size for particular topologies, that is $O\big(1/(1-\sigma_2)\big) = O(n^\beta)$ where $\beta = 2$ for a cycle, $\beta = 1$ for a grid and $\beta = 0$ for an expander, see for instance (Duchi et al., 2012). Naturally, more "connected" topologies have larger spectral gaps, and thus, smaller inverses.

Notation For $a, b \in \mathbb{R}$ we denote $a \vee b$ as the maximum between a and b and $a \wedge b$ the minimum. We say $a \simeq b$ if there exists a constant c independent of $n, m, M, (1-\sigma_2)^{-1}$ up-to logarithmic factors such that $a = cb$. Similarly we write $a \lesssim b$ if $a \leq bc$ and $a \gtrsim b$ if $a \geq cb$.

3. Main Results

This section presents the main results of this work. Section 3.1 provides the results under basic assumptions. Section 3.2 provides the results under more refined assumptions.

3.1. Basic Result

We begin by introducing the following assumption related to the feature map.

Assumption 1 *Let (Ω, π) be a probability space and define the feature map $\psi : X \times \Omega \to \mathbb{R}$ for all $x \in X$ such that (3) holds. Define the family of feature maps for $M > 0$*

$$\phi_M(x) := \frac{1}{\sqrt{M}} (\psi(x, \omega_1), \dots, \psi(x, \omega_M))$$

where $(\omega_j)_{j=1}^M \in \Omega$ are sampled independently from π.

The above assumption states that the feature map is made of M independent features $\psi(x, \omega_i)$ for $i = 1, \dots, M$. This is satisfied for a wide range of kernels, see for instance Appendix E of (Rudi & Rosasco, 2017). The next assumption introduces some regularity to the feature maps.

Assumption 2 *The function ψ is continuous and there exists $\kappa \geq 1$ such that $|\psi(x, \omega)| \leq \kappa$ for any $x \in X, \omega \in \Omega$.*

This implies that the kernel considered is bounded $|k(x, x')| \leq \kappa^2$ which is a common assumption in statistical learning theory (Cucker & Zhou, 2007; Steinwart & Christmann, 2008). The following assumption is related to the optimal predictor.

Assumption 3 *Let $\mathcal{H}$ be the RKHS with kernel k. Suppose there exists $f_{\mathcal{H}} \in \mathcal{H}$ such that $\mathcal{E}(f_{\mathcal{H}}) = \inf_{f \in \mathcal{H}} \mathcal{E}(f)$.*

It states that the optimal predictor is within the interior of $\mathcal{H}$. Moving beyond this assumption requires considering the non-attainable case, see for instance (Dieuleveut et al., 2016), which is left to future work. Finally, the following assumption is on the response moments.

Assumption 4 *For any $x \in X$*

$$\int y^{2\ell} d\rho(y|x) \leq \ell! B^\ell p, \quad \forall \ell \in \mathbb{N}$$

for constants $B \in (0, \infty)$ and $p \in (1, \infty)$, ρ_X-almost surely.

This assumption is satisfied if the response is bounded or generated from a model with independent zero mean Gaussian noise.

Given an estimator $\widehat{f}$, its excess risk is defined as $\mathcal{E}(\widehat{f}) - \mathcal{E}(f_{\mathcal{H}})$. Let the estimator held by agent $v \in V$ be denoted by $\widehat{f}_{t,v} = \langle \widehat{\omega}_{t,v}, \phi_M(\cdot) \rangle$, where $\widehat{\omega}_{t,v}$ is the output of Distributed Gradient Descent (4) for agent v. Given this basic setup, we state the prediction bound prescribed by our theory.

Theorem 1 (Basic Case) *Let $n, m, M \in \mathbb{N}_+$, $\delta \in (0, 1)$, $t \geq 4$, $\eta\kappa^2 \leq 1$ and $\eta \simeq 1$. Under assumptions 1 to 4, the following holds with high probability for any $v \in V$*

$$\mathcal{E}(\widehat{f}_{t+1,v}) - \mathcal{E}(f_{\mathcal{H}}) \lesssim \frac{1}{\sqrt{nm}}$$

when

$$m \gtrsim \frac{n^3}{(1-\sigma_2)^4}, \quad M \simeq \sqrt{nm}, \quad \text{and } t = \sqrt{nm}. \tag{5}$$

Theorem 1 demonstrates that Distributed Gradient Descent with Random Features achieves optimal statistical rates, in the minimax sense (Caponnetto & De Vito, 2007; Blanchard & Mücke, 2018), with respect to all nm samples when three conditions are met. The first $m \gtrsim n^3/(1-\sigma_2)^4$ ensures that the network errors, due to agents communicating locally on the network, are sufficiently small from the phenomena of concentration. The second $M \simeq \sqrt{nm}$ ensures that the agents have sufficiently many Random Features to control the kernel approximation. It aligns with the number required by single machine Gradient Descent with all nm samples (Carratino et al., 2018). Finally $t = \sqrt{nm}$ is the number of iterations required to trade off the bias and variance error terms. This is the number of iterations required by single machine Gradient Descent with all nm samples, and thus, due to considering a distributed algorithm, gives a linear speed-up in runtime. We now discuss the runtime and space complexity of Distributed Gradient Descent with Random Features when the covariates take values in $\mathbb{R}^D$ for some $D > 0$. Remark 1 in Appendix A shows how, with linear features, Random Features can yield communication savings when $D > M$.

Pre-processing + Space Complexity After a pre-processing step which costs $O(DMm) = O(Dm^{3/2}\sqrt{n})$, Distributed Gradient Descent has each agent store a $m \times M = m \times \sqrt{nm}$ matrix. Single machine Gradient Descent performs a $O(DMnm) = O(D(nm)^{3/2})$ pre-processing step and stores a $nm \times M = nm \times \sqrt{nm}$ matrix. Distributed Gradient Descent thus gives a linear order n improvement in pre-processing time and memory cost.

Time Complexity Suppose one gradient computation costs 1 unit of time and communicating with neighbours costs τ. Given sufficiently many samples $m \gtrsim n^3/(1-\sigma_2)^4$ then *Single Machine Iterations = Distributed Iterations* and the speed-up in runtime for Distributed Gradient Descent over single machine Gradient Descent is

$$\text{Speed-up} := \frac{\text{Single Machine Runtime}}{\text{Distributed Runtime}}$$
$$= \frac{\text{Single Machine Iteration Time}}{\text{Distributed Iteration Time}} \underbrace{\frac{\text{Single Machine Iters.}}{\text{Distributed Iters.}}}_{=1}$$
$$= \frac{nm}{m + \tau + M\text{Deg}(P)} \simeq n$$

where the final equality holds when the communication delay and cost of aggregating the neighbours solutions is bounded $\tau + M\text{Deg}(P) \lesssim m$. This observation demonstrates a linear speed-up in runtime can be achieved for *any* network topology. This is in contrast to results in decentralised consensus optimisation where the speed-up in runtime usually depends on the network topology, with a linear improvement only occurring for well connected topologies i.e. expander and complete, see for instance (Duchi et al., 2012; Scaman et al., 2017).

3.2. Refined Result

Let us introduce two standard statistical assumptions related to the underlying learning problem. With the marginal distribution on covariates $\rho_X(x) := \int_{\mathbb{R}} \rho(x,y)dy$ and the space of square integrable functions $L^2(X, \rho_X) = \{f : X \to \mathbb{R} : \|f\|_\rho^2 = \int |f|^2 d\rho_X < \infty\}$, let $L : L^2(X,\rho_X) \to L^2(X,\rho_X)$ be the integral operator defined for $x \in X$ as $Lf(x) = \int k(x,x')f(x')d\rho_X(x')$, $\forall f \in L^2(X,\rho_X)$. The above operator is symmetric and positive definite. The assumptions are then as follows.

Assumption 5 *For any $\lambda > 0$, define the effective dimension as $\mathcal{N}(\lambda) := \text{Tr}\left((L+\lambda I)^{-1}L\right)$, and assume there exists $Q > 0$ and $\gamma \in [0,1]$ such that $\mathcal{N}(\lambda) \leq Q^2\lambda^{-\gamma}$. Moreover, assume there exists $1 \geq r \geq 1/2$ and $g \in L^2(X,\rho_X)$ such that $f_{\mathcal{H}}(x) = (L^r g)(x)$.*

The above assumptions will allow more refined bounds on the Generalisation Error to be given. The quantity $\mathcal{N}(\lambda)$ is the effective dimension of the hypothesis space, and Assumption 5 holds for $\gamma > 0$ when the ith eigenvalue of L is of the order $i^{-1/\gamma}$, for instance. Meanwhile, the second condition for $1 \geq r \geq 1/2$ determines which subspace the optimal predictor is in. Here larger r indicates a smaller sub-space and a stronger condition. The refined result is then as follows.

Theorem 2 (Refined) *Let $n, m, M \in \mathbb{N}_+$, $\delta \in (0,1)$, $t \geq 2t^\star \geq 4$, $\eta\kappa^2 \leq 1$ and $\eta \simeq 1$. Under assumptions 1 to 5 with $r + \gamma > 1$, the following holds with high probability for any $v \in V$*

$$\mathcal{E}(\widehat{\omega}_{t+1,v}) - \mathcal{E}(f_{\mathcal{H}}) \lesssim (nm)^{\frac{-2r}{2r+\gamma}}$$

when we let $t^\star \simeq 1/(1-\sigma_2)$ and have

$$\underbrace{m \gtrsim \left((t^\star)^{\frac{(1+\gamma)(2r+\gamma)}{2(r+\gamma-1)}} n^{\frac{r+1}{r+\gamma-1}}\right) \vee \left((t^\star)^{2\vee(2r+\gamma)} n^{\frac{2r}{\gamma}}\right)}_{\textit{Sufficiently Many Samples}}$$
$$\underbrace{M \simeq (nm)^{\frac{1+\gamma(2r-1)}{2r+\gamma}}}_{\textit{Single Machine Random Features}} \qquad \underbrace{t = (nm)^{\frac{1}{2r+\gamma}}}_{\textit{Single Machine Iterations}}$$

Once again, the statistical rate achieved $(nm)^{-\frac{2r}{2r+\gamma}}$ is the minimax optimal rate with respect to all of the samples within the network (Caponnetto & De Vito, 2007), and both the number of Random Features as well as the number of iterations match the number required by single machine Gradient Descent when given *sufficiently many samples* m. When $r = 1/2$ and $\gamma = 1$ we recover the basic result given in Theorem 1, with the bounds now adapting to complexity

of the predictor as well as capacity through r and γ, respectively. In the low dimensional setting when $\gamma = 0$, we note our guarantees do not offer computational speed-ups over single machine Gradient Descent. While counter-intuitive, this observation aligns with (Richards & Rebeschini, 2019), which found the easier the problem (larger r, smaller γ) the more samples required to achieve a speed-up. This is due to network error concentrating at fixed rate of $1/m$ while the optimal statistical rate is $(nm)^{-\frac{2r}{2r+\gamma}}$. An open question is then how to modify the algorithm to exploit regularity and achieve a speed-up runtime, similar to how Leverage Score Sampling exploits additional regularity (Bach, 2013; Avron et al., 2017; Rudi et al., 2018; Li et al., 2019).

To provide insight into how the conditions in Theorem 2 arise, the following theorem gives the leading order error terms which contribute to the conditions in Theorem 2.

Theorem 3 (Leading Order Terms) *Let $n, m, M \in \mathbb{N}_+$, $\delta \in (0,1)$, $t \geq 2t^\star \geq 4$, $\eta\kappa^2 \leq 1$ and $\eta \simeq 1$. Under assumptions 1 to 5 with $r + \gamma > 1$, the following holds with high probability when $t^\star \simeq \frac{1}{1-\sigma_2}$ for any $v \in V$*

$$\mathcal{E}(\widehat{f}_{t+1,v}) - \mathcal{E}(f_{\mathcal{H}}) \lesssim \underbrace{\frac{\eta^\gamma}{m(1-\sigma_2)^\gamma} + \frac{(\eta t)^2(\eta t^\star)^{1+\gamma}}{m^2}}_{\textit{Network Error}}$$
$$\underbrace{\Big(\frac{\eta t}{M}+1\Big)\frac{(\eta t)^\gamma}{nm} + \frac{1}{M(\eta t)^{(1-\gamma)(2r-1)}} + \Big(\frac{1}{\eta t}\Big)^{2r}}_{\textit{Statistical Error}} + \textit{H.O.T.}$$

where H.O.T. denotes Higher Order Terms.

Theorem 3 decomposes the Generalisation Error into two terms. The *Statistical Error* matches the Generalisation Error of Gradient Descent with Random Features (Carratino et al., 2018) and consists of Sample Variance, Random Feature and Bias errors. The *Network Error* arises from tracking the difference between the Distributed Gradient Descent $\widehat{\omega}_{t+1,v}$ and single machine Gradient Descent iterates. The primary technical contribution of our work is in the analysis of this term, in particular, building on (Richards & Rebeschini, 2019) in two directions. Firstly, bounds are given in high probability instead of expectation. Secondly, we give a tighter analysis of the Residual Network Error, here denoted in the second half of the *Network Error* as $(\eta t)^2(\eta t^\star)^{1+\gamma}/m^2$. Previously this term was of the order $(\eta t)^{2+\gamma}/m^2$ and gave rise to the condition of $r + \gamma/2 \geq 1$, whereas we now require $r + \gamma \geq 1$. Our analysis can ensure it is decreasing with the step size η, and thus, be controlled by taking a smaller step size. While not explored in this work, we believe our approach would be useful for analysing the Stochastic Gradient Descent variant (Lin & Rosasco, 2017) where a smaller step size is often chosen.

4. Error Decomposition and Proof Sketch

In this section we give a more detailed error decomposition as well as a sketch of the proof. Section 4.1 gives the error decomposition into statistical and network terms. Section 4.2 decomposes the network term into a population and a residual part. Section 4.3 and 4.4 give sketch proofs for bounding the population and residual parts respectively.

4.1. Error Decomposition

We begin by introducing the iterates produced by a single machine Gradient Descent with nm samples as well as an auxiliary sequence associated to the population. Initialised at $\widehat{v}_1 = \widetilde{v}_1 = 0$, we define, for $t \geq 1$

$$\widehat{v}_{t+1} = \widehat{v}_t - \frac{\eta}{nm}\sum_{w\in V}\sum_{i=1}^{m}\big(\langle \widehat{v}_{t,w}, \phi_M(x_{i,w})\rangle - y_{i,w}\big)\phi_M(x_{i,w}),$$
$$\widetilde{v}_{t+1} = \widetilde{v}_t - \eta\int_X \big(\langle \widetilde{v}_t, \phi_M(x)\rangle - y\big)\phi_M(x)d\rho(x,y).$$

We work with functions in $L^2(X,\rho_X)$, thus we define $\widehat{g}_t = \langle \widehat{v}_t, \phi_M(\cdot)\rangle$, $\widetilde{g}_t = \langle \widetilde{v}_t, \phi_M(\cdot)\rangle$. Since the prediction error can be written in terms of the $L^2(X,\rho_X)$ as follows $\mathcal{E}(\widehat{f}_{t,v}) - \mathcal{E}(f_{\mathcal{H}}) = \|\widehat{f}_{t,v} - f_{\mathcal{H}}\|_\rho^2$ we have the decomposition $\widehat{f}_{t,v} - f_{\mathcal{H}} = \widehat{f}_{t,v} - \widehat{g}_t + \widehat{g}_t - f_{\mathcal{H}}$. The term $\widehat{g}_t - f_{\mathcal{H}}$ that we call the *Statistical Error* is studied within (Carratino et al., 2018). The primary contribution of our work is in the analysis of $\widehat{f}_{t,v} - \widehat{g}_t$ which we call the *Network Error*, and go on to describe in more detail next.

4.2. Network Error

To accurately describe the analysis for the network error we introduce some notation. Begin by defining the operator $S_M : \mathbb{R}^M \to L^2(X,\rho_X)$ so that $(S_M\omega)(\cdot) = \langle \omega, \phi_M(\cdot)\rangle$ as well as the covariance $C_M : \mathbb{R}^M \to \mathbb{R}^M$ defined as $C_M = S_M^\star S_M$, where $S_M^\star$ is the adjoint of S_M in $L^2(X,\rho_X)$. Utilising an isometry property (see (7) in the Appendix) we have for $\omega \in \mathbb{R}^M$ the following $\|S_M\omega\|_\rho = \|C_M^{1/2}\omega\|$, that is going from a norm in $L^2(X,\rho_X)$ to Euclidean norm. The empirical covariance operator of the covariates held by agent $v \in V$ is denoted $\widehat{C}_M^{(v)} : \mathbb{R}^M \to \mathbb{R}^M$. For $t \geq 1$ and a path $w_{t:1} = (w_t, w_{t-1}, \dots, w_1) \in V^t$ denote the collection of contractions

$$\Pi(w_{t:1}) = (I - \eta\widehat{C}_M^{(w_t)})(I - \eta\widehat{C}_M^{(w_{t-1})})\dots(I - \eta\widehat{C}_M^{(w_1)})$$

as well as the centered product $\Pi^\Delta(w_{t:1}) = \Pi(w_{t:1}) - (I - \eta C_M)^t$. For $w \in V$ $k \geq 1$ let $N_{k,w} \in \mathbb{R}^M$ denote a collection of zero mean random variables that are independent across agents $w \in V$ but not index $k \geq 1$.

For $v, w \in V$ and $s \geq 1$ define the difference $\Delta^s(v,w) := P_{vw}^s - \frac{1}{n}$, where we apply the power then index i.e.

$(P^s)_{vw} = P^s_{vw}$. For $w_{t:k} \in V^{t-k}$ denote the deviation along a path $\Delta(w_{t:k}) = P_{vw_{t:k}} - \frac{1}{n^{t-k}}$ where we have written the probability for a path $P_{vw_{t:k}} = P_{vw_t}P_{w_tw_{t-1}}\dots P_{w_{k+1}w_k}$.

Following (Richards & Rebeschini, 2019), center the distributed $\omega_{t+1,v}$ and the single machine iterates $\widehat{v}_{t+1}$ around the population iterates $\widetilde{v}_t$. Apply the isometry property to $\|\widehat{f}_{t,v} - \widehat{g}_t\|_\rho = \|C_M^{1/2}(\widehat{\omega}_{t+1,v} - \widehat{v}_t)\|$ and following the steps in Appendix D.1 we arrive at

$$\|C_M^{1/2}(\widehat{\omega}_{t+1,v} - \widehat{v}_{t+1})\| \le \underbrace{\sum_{k=1}^{t}\eta\sum_{w\in V}|\Delta^{t-k}(v,w)|\|C_M^{1/2}(I-\eta C_M)^{t-k}N_{k,w}\|}_{\text{Population Network Error}} + \underbrace{\sum_{k=1}^{t}\eta\Big\|\sum_{w_{t:k}\in V^{t-k+1}}\Delta(w_{t:k})C_M^{1/2}\Pi^\Delta(w_{t:k+1})N_{k,w_k}\Big\|}_{\text{Residual Network Error}}.$$

The two terms above can be associated to the two terms in the network error of Theorem 3, with the Population Network Error decreasing as $1/m$ and the Residual Network Error as $1/m^2$. We now analyse each of these terms separately.

4.3. Network Error: Population

Our contribution for analysing the *Population Network Error* is to give bounds it in high probability, where as (Richards & Rebeschini, 2019) only gave bounds in expectation. Choosing some $t \ge 2t^\star \ge 2$ and splitting the series at $k = t - t^\star$ we are left with two terms. For $1 \le k \le t - t^\star$ we utilise that the sum over the difference $|\Delta^s(v,w)|$ can be written in terms of euclidean ℓ_1 norm and this is bounded by the second largest eigenvalue of P in absolute value i.e. $\sum_{w\in V}|\Delta^{t-k}(v,w)| = \|e_v^\top P^{t-k} - \frac{1}{n}\mathbf{1}\|_1 \le \sqrt{n}\sigma_2^{t-k} \le \sqrt{n}\sigma_2^{t^\star}$, where e_v is the standard basis vector in $\mathbb{R}^n$ with a 1 aligning with agent $v \in V$ and $\mathbf{1}$ is a vector of all 1's. Meanwhile for $t \ge k \ge t - t^\star$, we follow (Richards & Rebeschini, 2019) and utilise the contraction of the gradient updates i.e. $C_M^{1/2}(I - \eta C_M)^{t-k}$ alongside that N_{k,w_k} is an average of m i.i.d. random variables, and thus, concentrate at $1/\sqrt{m}$ in high probability. This leads to the bound in high probability

$$\text{Population Network Error} \lesssim \underbrace{\frac{\sqrt{n}\sigma_2^{t^\star}t}{\sqrt{m}}}_{\text{Well Mixed Terms}} + \underbrace{\frac{(\eta t^\star)^{\gamma/2}}{\sqrt{m}}}_{\text{Poorly Mixed Terms}}.$$

The first term *Well Mixed*, decays exponentially with the second largest eigenvalue of P in absolute value, and represents the information from past iterates that has now fully propagated around the network. The term *Poorly Mixed* represents error from the most recent iterates that is yet to fully propagate through the network. It grows at the rate $(t^\star)^{\gamma/2}$ due to utilising the contractions of the gradients as well as the assumptions 5. The quantity $t^\star$ is now chosen to trade off these terms. Note by writing $\sigma_2^{t^\star} = e^{-t^\star\log(1/\sigma_2)}$ that, up to logarithmic factors, the first can be made small by taking $t^\star \gtrsim \frac{1}{1-\sigma_2} \ge \frac{1}{-\log(\sigma_2)}$.

4.4. Network Error: Residual

The primary technical contribution of our work is in the analysis of this term. The analysis builds on insights from (Richards & Rebeschini, 2019), specifically that $\Pi^\Delta(w_{t:1})$ is a product of empirical operators minus the population, and thus, can be written in terms of the differences $\widehat{C}_M^{(w)} - C_M$ which concentrate at $1/\sqrt{m}$. Specifically, for $N \in \mathbb{R}^M$, the bound within (Richards & Rebeschini, 2019) was of the following order with high probability for any $w_{t:1} \in V^t$

$$\|C_M^{1/2}\Pi^\Delta(w_{t:1})N\| \lesssim \|N\|\frac{(\eta t)^{\gamma/2}}{\sqrt{m}}. \tag{6}$$

The bound for Residual Network Error within (Richards & Rebeschini, 2019) is arrived at by applying triangle inequality over the series $\sum_{w_{t:k}\in V^{t-k+1}}$, plugging in (6) for $\|C_M^{1/2}\Pi^\Delta(w_{t:k+1})N_{k,w_k}\|$ alongside $\|N_{k,w_k}\| \lesssim 1/\sqrt{m}$ see Lemma 7 in Appendix. Summing over $1 \le k \le t$ yields the bound of order $(\eta t)^{1+\gamma/2}/m$ in high probability. The two key insights of our analysis are as follows. Firstly, noting that the error for bounding the contraction $\Pi^\Delta(w_{t:1})$ grows with the length of the path, and as such, we should aim to apply the bound (6) to short paths. Secondly, note for $N \in \mathbb{R}^M$ quantities of the form $\|C_M^{1/2}\sum_{w_{t:1}\in V^t}\Delta(w_{t:1})\Pi^\Delta(w_{t:1})N\|$ concentrate quickly (Lemma 13 in Appendix).

To apply the insights outlined previously, we decompose the deviation $\Pi^\Delta(w_{t:2})$ into two terms that only replace the final $t^\star$ operators with the population, that is

$$\Pi^\Delta(w_{t:2}) = \Pi(w_{t:t^\star+2})\Pi^\Delta(w_{t^\star+1:1}) + \Pi^\Delta(w_{t:t^\star+2})(I-\eta C_M)^{t^\star}.$$

Plugging in the above then yields, for the case $k = 1$,

$$\begin{aligned}&\sum_{w_{t:1}\in V^t}\Delta(w_{t:1})C_M^{1/2}\Pi^\Delta(w_{t:2})N_{k,w_1}\\ &= \sum_{w_{t:1}\in V^t}\Delta(w_{t:1})C_M^{1/2}\Pi(w_{t:t^\star+2})\underbrace{\Pi^\Delta(w_{t^\star+1:1})}_{t^\star\text{contraction}}N_{k,w_1}\\ &+\sum_{w_{t:1}\in V^t}\Delta(w_{t:1})\underbrace{C_M^{1/2}\Pi^\Delta(w_{t:t^\star+2})(I-\eta C_M)^{t^\star}N_{k,w_1}}_{\text{Independent of } w_{t^\star+1:1}}\end{aligned}$$

Note that the first term above only contains a contraction $\Pi^\Delta(w_{t^\star+1:1})$ of length $t^\star$, and as such, when applying a variant of (6) will only grow at length $(\eta t^\star)^{(1+\gamma)/2}/\sqrt{m}$.

When summing over $1 \leq k \leq t$ this will result in the leading order term for the residual error of $(\eta t)(\eta t^\star)^{(1+\gamma)/2}/m$. For the second term, note the highlighted section is independent of the final $t^\star$ steps of the path $w_{t:1}$, namely $w_{t^\star+1:1}$. Therefore we can sum the deviation $\Delta(w_{t:1})$ over path $w_{t^\star+1:1}$ and, if $t^\star \gtrsim \frac{1}{1-\sigma_2}$, replace N_{k,w_1} by the average $\frac{1}{n}\sum_{w\in V} N_{k,w}$. This has impact of decoupling the summation over the remainder of the path $w_{t:t^\star}$ allowing the second insight from previously to be used. For details on this step we point the reader to Appendix Section D.1.

5. Experiments

For our experiments we consider subsets of the SUSY data set (Baldi et al., 2014), as well as single machine and Distributed Gradient Descent with a fixed step size $\eta = 1$. Cycle and grid network topologies are studied, with the matrix P being a simple random walk. Random Fourier Features are used $\psi(x, \omega) = \cos(\xi \times w^\top x + q)$, with $\omega := (w, q)$, w sampled according to the normal distribution, q sampled uniformly at random between 0 and 2π, and ξ is a tuning parameter associated to the bandwidth (fixed to $\xi = 10^{-1/2}$). For any given sample size, topology or network size we repeated the experiment 5 times. Test size of 10^4 was used and classification error is minimum over iterations and maximum over agents i.e. $\min_t \max_{v\in V} \mathcal{E}_{\text{Approx}}(\widehat{\omega}_{t,v})$, where $\mathcal{E}_{\text{Approx}}$ is approximated test error. With the response of the data being either 1 or 0 and the predicted response $\widehat{y}$, the predicted classification is the indicator function of $\widehat{y} > 1/2$. The classification error is the proportion of mis-classified samples.

We begin by investigating the number of Random Features required with Distributed Gradient Descent to match the single machine performance. Looking to Figure 1, observe that for a grid topology, as well as small cycles ($n = 9, 25$), that the classification error aligns with a single machine beyond approximately $\sqrt{nm}$ Random Features. For larger more poorly connected topologies, in particular a cycle with $n = 49$ agents, we see that the error does not fully decrease down that of single machine Gradient Descent.

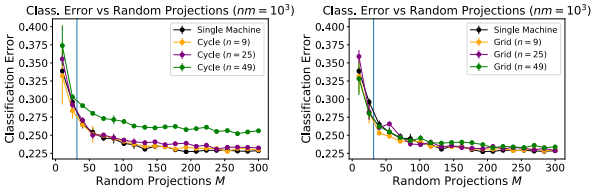


Figure 1. Classification Error (if y and $\widehat{y}$ are the true and predicted response respectively, error calculated is 0-1 loss) against number of Random Features M, with total sample size and maximum number of iterations $t = nm = 10^3$. Vertical line in plots indicates $\sqrt{nm}$. *Left*: Cycle topology, *Right*: Grid Topology.

Our theory predicts that the sub-optimality of more poorly connected networks decreases as the number of samples held by each agent increases. To investigate this, we repeat the above experiment for cycles and grids of sizes $n = 25, 49, 100$ while varying the dataset size. Looking to Figure 2, we see that approximately $nm \approx 10^3$ samples are sufficient for a cycle topology of size $n = 49$ to align with a single machine, meanwhile 10^4 samples are required for a larger $n = 100$ cycle. For a grid we see a similar phenomena, although with fewer samples required due to being better connected topology.

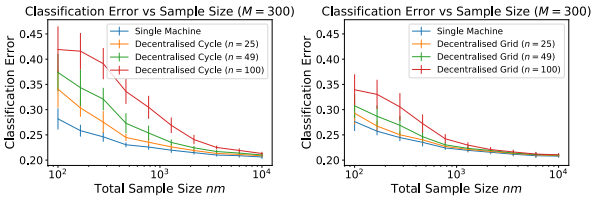


Figure 2. Plots of Classification Error (computed as in Figure 1) against total number of samples nm, with $M = 300$. Run for at most $t = 10^4$ iterations, each point is an average of 20 sub-subsets of the SUSY, which Distributed Gradient Descent with Random Features is run on 5 times.

Our theory predicts that, given sufficiently many samples, the number of iterations for any network topology scales as those of single machine Gradient Descent. We look to Figure 3 where the number of iterations required to achieve the minimum classification error (optimal stopping time) is plotted against the sample size. Observe that beyond approximately 10^3 samples both grid and cycles of sizes $n = 25, 49, 100$ have iterates that scale at the same order as a single machine. Observe that the number of iterations required by the grid and cycle topologies is initially decreasing with the sample size up to 10^3. While not supported by our theory for a constant step size, this suggests quantities held by agents become similar as agents hold more data, reducing the need for additional iterations in order to propagate information around the network. An analysis of this phenomena we leave to future work.

6. Conclusion

In this work we considered the performance of Distributed Gradient Descent with Random Features on the Generalisation Error, this being different from previous works which focused on training loss. Our analysis allowed us to understand the role of different parameters on the Generalisation error, and, when agents have sufficiently many samples with respect to the network size, achieve a linear speed-up in runtime time for any network topology.

Moving forward, it would be natural to extend our analysis to stochastic gradients (Lin & Rosasco, 2017) or stochastic

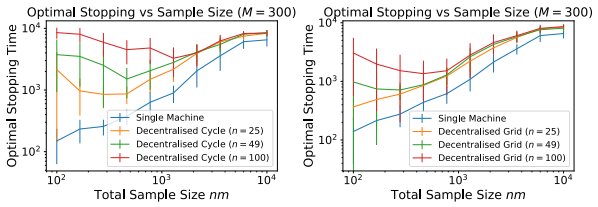


Figure 3. Optimal Stopping Time (Number of iterations required) against sample size nm ($\log - \log$ axis), with $M = 300$. *Left*: Cycle Topology, *Right*: Grid topology. Each point is averaged over 20 sub-subsets of the SUSY. Distributed Gradient Descent with Random Features was repeated 5 times, with at most 10^4 iterations.

communication at each iteration (Shah, 2009).

Acknowledgements

D.R. is supported by the EPSRC and MRC through the OxWaSP CDT programme (EP/L016710/1). Part of this work has been carried out at the Machine Learning Genoa (MaLGa) center, Università di Genova (IT). L.R. acknowledges the financial support of the European Research Council (grant SLING 819789), the AFOSR projects FA9550-17-1-0390 and BAA-AFRL-AFOSR-2016-0007 (European Office of Aerospace Research and Development), and the EU H2020-MSCA-RISE project NoMADS - DLV-777826.

References

Aronszajn, N. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

Avron, H., Kapralov, M., Musco, C., Musco, C., Velingker, A., and Zandieh, A. Random Fourier features for kernel ridge regression: Approximation bounds and statistical guarantees. In *34th International Conference on Machine Learning*, pp. 253–262. PMLR, 2017.

Bach, F. Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory*, pp. 185–209, 2013.

Baldi, P., Sadowski, P., and Whiteson, D. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5:4308, 2014.

Blanchard, G. and Mücke, N. Optimal rates for regularization of statistical inverse learning problems. *Foundations of Computational Mathematics*, 18(4):971–1013, 2018.

Bouboulis, P., Chouvardas, S., and Theodoridis, S. Online distributed learning over networks in rkh spaces using random fourier features. *IEEE Transactions on Signal Processing*, 66(7):1920–1932, 2017.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011.

Caponnetto, A. and De Vito, E. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

Carratino, L., Rudi, A., and Rosasco, L. Learning with sgd and random features. In *Advances in Neural Information Processing Systems*, pp. 10192–10203, 2018.

Chouvardas, S. and Draief, M. A diffusion kernel lms algorithm for nonlinear adaptive networks. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4164–4168. IEEE, 2016.

Cucker, F. and Zhou, D. X. *Learning theory: an approximation theory viewpoint*, volume 24. Cambridge University Press, 2007.

Dieuleveut, A., Bach, F., et al. Nonparametric stochastic approximation with large step-sizes. *The Annals of Statistics*, 44(4):1363–1399, 2016.

Dobriban, E. and Sheng, Y. Wonder: Weighted one-shot distributed ridge regression in high dimensions. *Journal of Machine Learning Research*, 21(66):1–52, 2020.

Duchi, J. C., Agarwal, A., and Wainwright, M. J. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic Control*, 57(3):592–606, 2012.

Forero, P. A., Cano, A., and Giannakis, G. B. Consensus-based distributed support vector machines. *Journal of Machine Learning Research*, 11(May):1663–1707, 2010.

Gao, W., Chen, J., Richard, C., and Huang, J. Diffusion adaptation over networks with kernel least-mean-square. In *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pp. 217–220. IEEE, 2015.

Guo, Z.-C., Lin, S.-B., and Zhou, D.-X. Learning theory of distributed spectral algorithms. *Inverse Problems*, 33(7): 074009, 2017.

Jakovetić, D., Moura, J. M., and Xavier, J. Linear convergence rate of a class of distributed augmented lagrangian algorithms. *IEEE Transactions on Automatic Control*, 60 (4):922–936, 2015.

Johansson, B., Rabi, M., and Johansson, M. A simple peer-to-peer algorithm for distributed optimization in sensor networks. In *Decision and Control, 2007 46th IEEE Conference on*, pp. 4705–4710. IEEE, 2007.

Johansson, B., Rabi, M., and Johansson, M. A randomized incremental subgradient method for distributed optimization in networked systems. *SIAM Journal on Optimization*, 20(3):1157–1170, 2009.

Koppel, A., Paternain, S., Richard, C., and Ribeiro, A. Decentralized online learning with kernels. *IEEE Transactions on Signal Processing*, 66(12):3240–3255, 2018.

Le, Q., Sarlós, T., and Smola, A. Fastfood-computing hilbert space expansions in loglinear time. In *International Conference on Machine Learning*, pp. 244–252, 2013.

Li, Z., Ton, J.-F., Oglic, D., and Sejdinovic, D. Towards a unified analysis of random fourier features. In *International Conference on Machine Learning*, pp. 3905–3914, 2019.

Lin, J. and Cevher, V. Optimal convergence for distributed learning with stochastic gradient methods and spectral-regularization algorithms. *arXiv preprint arXiv:1801.07226*, 2018.

Lin, J. and Rosasco, L. Optimal rates for multi-pass stochastic gradient methods. *Journal of Machine Learning Research*, 18(97):1–47, 2017.

Lin, S.-B., Guo, X., and Zhou, D.-X. Distributed learning with regularized least squares. *The Journal of Machine Learning Research*, 18(1):3202–3232, 2017.

Lobel, I. and Ozdaglar, A. Distributed subgradient methods for convex optimization over random networks. *IEEE Transactions on Automatic Control*, 56(6):1291–1306, 2011.

Matei, I. and Baras, J. S. Performance evaluation of the consensus-based distributed subgradient method under random communication topologies. *IEEE Journal of Selected Topics in Signal Processing*, 5(4):754–771, 2011.

Mitra, R. and Bhatia, V. The diffusion-klms algorithm. In *2014 International Conference on Information Technology*, pp. 256–259. IEEE, 2014.

Mokhtari, A. and Ribeiro, A. Dsa: Decentralized double stochastic averaging gradient algorithm. *Journal of Machine Learning Research*, 17(61):1–35, 2016.

Mücke, N. and Blanchard, G. Parallelizing spectrally regularized kernel algorithms. *The Journal of Machine Learning Research*, 19(1):1069–1097, 2018.

Nedic, A. and Ozdaglar, A. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

Nedić, A., Olshevsky, A., Ozdaglar, A., and Tsitsiklis, J. N. On distributed averaging algorithms and quantization effects. *IEEE Transactions on Automatic Control*, 54(11): 2506–2517, 2009.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pp. 1177–1184, 2008.

Reed, M. *Methods of modern mathematical physics: Functional analysis*. Elsevier, 2012.

Richards, D. and Patrick, R. Graph-dependent implicit regularisation for distributed stochastic subgradient descent. *Journal of Machine Learning Research*, 21(2020):1–44, 2020.

Richards, D. and Rebeschini, P. Optimal statistical rates for decentralised non-parametric regression with linear speed-up. In *Advances in Neural Information Processing Systems*, pp. 1216–1227, 2019.

Rudi, A. and Rosasco, L. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems*, pp. 3215–3225, 2017.

Rudi, A., Calandriello, D., Carratino, L., and Rosasco, L. On fast leverage score sampling and optimal learning. In *Advances in Neural Information Processing Systems*, pp. 5672–5682, 2018.

Scaman, K., Bach, F., Bubeck, S., Lee, Y. T., and Massoulié, L. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *34th International Conference on Machine Learning*, pp. 3027–3036. PMLR, 2017.

Schölkopf, B., Herbrich, R., and Smola, A. J. A generalized representer theorem. In *International conference on computational learning theory*, pp. 416–426. Springer, 2001.

Shah, D. Gossip algorithms. *Foundations and Trends® in Networking*, 3(1):1–125, 2009.

Shi, W., Ling, Q., Yuan, K., Wu, G., and Yin, W. On the linear convergence of the admm in decentralized consensus optimization. *IEEE Trans. Signal Processing*, 62(7): 1750–1761, 2014.

Shi, W., Ling, Q., Wu, G., and Yin, W. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

Steinwart, I. and Christmann, A. *Support vector machines*. Springer Science & Business Media, 2008.

Tsitsiklis, J., Bertsekas, D., and Athans, M. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE transactions on automatic control*, 31(9):803–812, 1986.

Tsitsiklis, J. N. Problems in decentralized decision making and computation. Technical report, Massachusetts Inst Of Tech Cambridge Lab For Information And Decision Systems, 1984.

Xu, P., Wang, Y., Chen, X., and Zhi, T. Coke: Communication-censored kernel learning for decentralized non-parametric learning. *arXiv preprint arXiv:2001.10133*, 2020.

Yu, F. X. X., Suresh, A. T., Choromanski, K. M., Holtmann-Rice, D. N., and Kumar, S. Orthogonal random features. In *Advances in Neural Information Processing Systems*, pp. 1975–1983, 2016.

Zhang, J., May, A., Dao, T., and Ré, C. Low-precision random fourier features for memory-constrained kernel approximation. *Proceedings of machine learning research*, 89:1264, 2019.

Zhang, Y., Duchi, J., and Wainwright, M. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *The Journal of Machine Learning Research*, 16(1):3299–3340, 2015.

A. Remarks

In this section we give a number of remarks relating to content within the main body of the paper.

Remark 1 (Sketching and Communication Savings) *We highlight that the Random Feature framework considered also incorporates a number of sketching techniques. For instance, when $\psi(x, \omega) = x^\top \omega$ where $\omega \sim \mathcal{N}(0, I)$ and the associated kernel is simply linear as $\mathbf{E}[\psi(x, \omega)\psi(x', \omega)]\mathbf{E}[x^\top \omega\omega^\top x] = x^\top \mathbf{E}[\omega\omega^\top]x = x^\top x'$. The case $M < D$ then represents a simple setting in which communication savings can be achieved, as agents in this case would only need to communicate an M dimensional vector instead of D. A natural future direction would be to investigate whether there exists particular sketches/Random Features tailored to the objective of communication savings, in a similar manner to Orthogonal Random Features (Yu et al., 2016), Fast Food (Le et al., 2013) or Low-precision Random Features (Zhang et al., 2019). Although, as noted in (Carratino et al., 2018), some of these methods sample the features in a correlated manner, and thus, do not fit within the assumptions of this work.*

Remark 2 (Previous Literature Decentralised Kernel Methods) *This remark highlights two previous works for Decentralised Kernel Methods. The work (Forero et al., 2010) considers decentralised Support Vector Machines with potentially high-dimensional finite feature spaces that could approximate a non-linear kernel. They develop a variant of the Alternating Direction Method of Multiplers (ADMM) to target the augmented optimisation problem. In this case, the high-dimensional constraints across the agents are approximated so the agents local estimated functions are equal on a subset of chosen points. Meanwhile (Koppel et al., 2018) consider online stochastic optimisation with penalisation between neighbouring agents. The penalisation introduced is an expectation with respect to a newly sampled data point and not in the norm of the Reproducing Kernel Hilbert Space. In both of these cases, the original optimisation problem is altered to facilitate a decentralised algorithm, but no guarantee is given on how these approximation impact statistical performance.*

Remark 3 (Concurrent Work) *The concurrent work (Xu et al., 2020) consider the homogeneous setting where a network of agents have data from the same distribution and wish to learn a function within a RKHS that performs well on unseen data. The consensus optimisation formulation of the single machine explicitly penalised kernel learning problem is considered, and the challenges of decentralised kernel learning (as described in Section 2.1 in the main body of the manuscript) are overcome by utilising Random Fourier Features. An ADMM method is developed to solve the consensus optimisation problem, and, provided hyper-parameters are tuned appropriately, optimisation guarantees are given. Due to considering the consensus optimisation formulation of a single machine penalised problem, the Generalisation Error is decoupled from the Optimisation Error. Therefore, while optimisation results for ADMM applied to consensus optimisation objectives (Shi et al., 2014) are applied, the statistical setting is not leveraged to achieve speed-ups. It is then not clear how the network connectivity, number of samples held by agents and finer statistical assumptions (source and capacity) impacts either generalisation or optimisation performance. This is in contrast to our work, where we directly study the Generalisation Error of Distributed Gradient Descent with Implicit Regularisation, and show how the number of samples held by agents, network topology, step size and number of iterations can impact Generalisation Error.*

B. Analysis Setup

This section provides the setup for the analysis. We adopt the notation of (Carratino et al., 2018), which is included here for completeness. Section B.1 introduces additional auxiliary quantities required for the analysis. Section B.2 introduces notation for the operators required for the analysis. Section B.3 introduces the error decomposition.

B.1. Additional Auxiliary Sequences

We begin by introducing some auxiliary sequences that will be useful in the analysis. Begin by defining $\{v_t\}_{t\geq 1}$ initialised at $v_1 = 0$ and updated for $t \geq 1$ and updated

$$v_{t+1} = v_t - \eta \int_X \big(\langle v_t, \phi_M(x)\rangle - f_{\mathcal{H}}(x)\big)\phi_M(x) d\rho_X(x)$$

Further for $\lambda > 0$ let

$$\widetilde{u}_\lambda = \text{argmin}_{u\in\mathbb{R}^M} \int_X \big(\langle u, \phi_M(x)\rangle - f_{\mathcal{H}}(x)\big)^2 d\rho_X(x) + \lambda\|u\|^2,$$
$$u_\lambda = \text{argmin}_{u\in\mathcal{F}} \int_X \big(\langle u, \phi(x)\rangle - y)^2 d\rho(x,y) + \lambda\|u\|^2,$$

where $(\mathcal{F}, \phi)$ are feature space and feature map associated to the kernel k. As described previously, it will be useful to work with functions in $L^2(X, \rho_X)$, therefore define the functions

$$g_t = \langle v_t, \phi_M(\cdot)\rangle, \quad \widetilde{g}_\lambda = \langle \widetilde{u}_\lambda, \phi_M(\cdot)\rangle, \quad g_\lambda = \langle u_\lambda, \phi(\cdot)\rangle.$$

The quantities introduced here in this section will be useful in analysing the *Statistical Error* term.

B.2. Notation

Let $\mathcal{F}$ be the feature space corresponding to the kernel k given by Assumption 2.

Given $\phi : X \to \mathcal{F}$ (feature map), we define the operator $S : \mathcal{F} \to L^2(X, \rho_X)$ as

$$(S\omega)(\cdot) = \langle \omega, \phi(\cdot)\rangle_{\mathcal{F}}, \quad \forall \omega \in \mathcal{F}.$$

If $S^\star$ is the adjoint operator of S, we let $C : \mathcal{F} \to \mathcal{F}$ be the linear operator $C = S^\star S$, which can be written as

$$C = \int_X \phi(x) \otimes \phi(x) d\rho_X(x).$$

We also define the linear operator $L : L^2(X, \rho_X) \to L^2(X, \rho_X)$ such that $L = SS^\star$, that can be represented as

$$(Lf)(\cdot) = \int_X \langle \phi(x), \phi(\cdot)\rangle_{\mathcal{F}} f(x) d\rho_X(x), \quad \forall f \in L^2(X, \rho_X).$$

We now define the analog of the previous operators where we use the feature map ϕ_M instead of ϕ. We have $S_M : \mathbb{R}^M \to L^2(X, \rho_X)$ defined as

$$(S_M v)(\cdot) = \langle v, \phi_M(\cdot)\rangle_{\mathbb{R}^M}, \quad \forall v \in \mathbb{R}^M$$

together with $C_M : \mathbb{R}^M \to \mathbb{R}^M$and $L_M : L^2(X, \rho_X) \to L^2(X, \rho_X)$ defined as $C_M = S_M^\star S_M$ and $L_M = S_M S_M^\star$ respectively. For $v \in \mathbb{R}^M$ note we have the equality

$$\begin{aligned}
\|S_M v\|_\rho^2 &= \int_X \langle v, \phi_M(x)\rangle^2 d\rho_X(x) \\
&= \int_X v^\top \phi_M(x) \otimes \phi_M(x) v d\rho_X(x) \\
&= v^\top C_M v \\
&= \|C_M^{1/2} v\|^2
\end{aligned} \tag{7}$$

where we have denoted the standard Euclidean norm as $\|\cdot\|$. Define the empirical counterpart of the previous operators for each agent. For each agent $v \in V$ define the operator $\widehat{S}_M^{(v)} : \mathbb{R}^M \to \mathbb{R}^m$ as

$$\widehat{S}_M^{(v)\top} = \frac{1}{\sqrt{m}}(\phi_M(x_{1,v}), \ldots, \phi_M(x_{m,v})),$$

and with $\widehat{C}_M^{(v)} : \mathbb{R}^M \to \mathbb{R}^M$ and $\widehat{L}_M^{(v)} : \mathbb{R}^m \to \mathbb{R}^m$ are defined as $\widehat{C}_M^{(v)} = \widehat{S}_M^{(v)\top}\widehat{S}_M^{(v)}$ and $\widehat{L}_M^{(v)} = \widehat{S}_M^{(v)}\widehat{S}_M^{(v)\top}$ respectively. Moreover, define the empirical operators associated to all of the samples held by agents in the network. To do so index the agents in V between 1 and n, so $x_{i,j}$ is the ith data point held by agent j. Then, define the operator $\widehat{S}_M : \mathbb{R}^M \to \mathbb{R}^{nm}$ as

$$\begin{aligned}
\widehat{S}_M^\top &= \frac{1}{\sqrt{nm}}(\phi_M(x_{1,1}), \ldots, \phi_M(x_{m,1}), \phi_M(x_{1,2}), \ldots, \phi_M(x_{m,2}), \ldots, \phi_M(x_{1,n}), \ldots, \phi_M(x_{m,n})) \\
&= \frac{1}{\sqrt{n}}(\widehat{S}_M^{(1)\top}, \ldots, \widehat{S}_M^{(n)\top})
\end{aligned}$$

and with $\widehat{C}_M : \mathbb{R}^M \to \mathbb{R}^M$ and $\widehat{L}_M : \mathbb{R}^{nm} \to \mathbb{R}^{nm}$ are defined as $\widehat{C}_M = \widehat{S}_M^\top \widehat{S}_M$ and $\widehat{L}_M = \widehat{S}_M \widehat{S}_M^\top$ respectively. From the above it is clear that we have $\widehat{C}_M = \frac{1}{n}\sum_{w\in V} \widehat{S}_M^{(w)\top}\widehat{S}_M^{(w)} = \frac{1}{n}\sum_{w\in V} C_M^{(w)}$. For some number $\lambda > 0$ we let the operator plus the identity times λ be denoted $L_\lambda = L + \lambda I$, and similarly for $\widehat{L}_\lambda$, as well as $C_{M,\lambda} = C_M + \lambda I$ and $\widehat{C}_{M,\lambda}$.

Remark 4 *Let $P : L^2(X, \rho_X) \to L^2(X, \rho_X)$ be the projection operator whose range is the closure of the range of L. Let $f_\rho : X \to \mathbb{R}$ be defined as*

$$f_\rho(x) = \int y d\rho(y|x).$$

If there exists $f_\mathcal{H} \in \mathcal{H}$ such that

$$\inf_{f\in\mathcal{H}} \mathcal{E}(f) = \mathcal{E}(f_\mathcal{H})$$

then

$$Pf_\rho = Sf_\mathcal{H}.$$

or equivalently, there exists $g \in L^2(X, \rho_X)$ such that

$$Pf_\rho = L^{1/2} g.$$

In particular, we have $R := \|f_\mathcal{H}\|_\mathcal{H} = \|g\|_{L^2(X,\rho_X)}$. The above condition is commonly relaxed in approximation theory as

$$Pf_\rho = L^r g$$

with $1/2 \le r \le 1$.

With the operators introduced above and the above remark, we can rewrite the auxiliary objects respectively as

$$\begin{aligned}
\widehat{v}_1 = 0; \quad \widehat{v}_{t+1} &= (I - \eta\widehat{C}_M)\widehat{v}_t + \eta\widehat{S}_M^\top \widehat{y} \\
\widetilde{v}_1 = 0; \quad \widetilde{v}_{t+1} &= (I - \eta C_M)\widetilde{v}_t + \eta S_M^\star f_\rho \\
v_1 = 0; \quad v_{t+1} &= (I - \eta C_M)v_t + \eta S_M^\star P f_\rho
\end{aligned}$$

where the vector of all nm responses are $\widehat{y}^\top = (nm)^{-1/2}(y_{1,1}, \ldots, y_{1,m}, y_{2,m}, \ldots, y_{n,m}) = (n)^{-1/2}(\widehat{y}_1, \ldots, \widehat{y}_n)$, and each agents responses are, for $i = 1, \ldots, n$, denoted $\widehat{y}_v = (m)^{-1/2}(y_{i,1} \ldots, y_{i,m})$. We then denote

$$\begin{aligned}
\widetilde{u}_\lambda &= S_M^\star L_{M,\lambda}^{-1} P f_\rho \\
u_\lambda &= S^\star L_\lambda^{-1} P f_\rho.
\end{aligned}$$

Inductively the three sequences can be written as

$$\begin{aligned}
\widehat{v}_{t+1} &= \sum_{k=1}^{t} \eta(I - \eta\widehat{C}_M)^{t-k}\widehat{S}_M^\top \widehat{y} \\
\widetilde{v}_{t+1} &= \sum_{k=1}^{t} \eta(I - \eta C_M)^{t-k} S_M^\star f_\rho \\
v_{t+1} &= \sum_{k=1}^{t} \eta(I - \eta C_M)^{t-k} S_M^\star P f_\rho
\end{aligned}$$

B.3. Error Decomposition

We can now write the deviation $\widehat{f}_{t+1,v} - f_\mathcal{H}$ using the operators

$$\widehat{f}_{t+1,v} - f_\mathcal{H} = \underbrace{S_M\widehat{\omega}_{t+1,v} - S_M\widehat{v}_t}_{\text{Network Error}} + \underbrace{S_M\widehat{v}_t - Pf_\rho}_{\text{Statistical Error}} \tag{8}$$

where the first term aligns with the network error and the second with the statistical error. Each of these will be analysed in it own section.

C. Statistical Error

In this section we summarise the analysis for the Statistical Error which has been conducted within (Carratino et al., 2018). Here we provided the proof for completeness. Firstly, we further decompose the statistical error into the following terms

$$\|S_M\widehat{v}_{t+1} - Pf_\rho\|_\rho \leq \underbrace{\|S_M\widehat{v}_{t+1} - S_M\widetilde{v}_{t+1} + S_M\widetilde{v}_{t+1} - S_M v_t\|_\rho}_{\text{Sample Error}} + \underbrace{\|S_M v_{t+1} - L_M L_{M,\lambda}^{-1} Pf_\rho\|_\rho}_{\text{Gradient Descent and Ridge Regression}} \tag{9}$$
$$+ \quad \underbrace{\|L_M L_{M,\lambda}^{-1} Pf_\rho - LL_\lambda^{-1} Pf_\rho\|_\rho}_{\text{Random Features Error}} + \underbrace{\|LL_\lambda^{-1} Pf_\rho - Pf_\rho\|_\rho}_{\text{Bias}}$$

Each of the terms have been labelled to help clarity. The first term, *sample error* includes the difference between the empirical iterations with sampled data $\widehat{v}_t$, as well as iterates under the population measure v_t. The second term *Gradient Descent and Ridge Regression* is the difference between the population variants of the Gradient Descent v_t and ridge regression $L_M L_{M,\lambda}^{-1} Pf_\rho$ solutions. The third term *Random Feature Error* accounts for the error introduced from using Random Features. Finally the *Bias* term accounts for the bias introduced due to the regularisation. Each of these terms will be bounded within their own sub-section, except the *Bias* term which will be bounded when bounds for all of the terms are brought together.

The remainder of this section is then as follows. Section C.1, C.2 and C.3 give the analysis for the Sample Error, Gradient Descent and Ridge Regression and Random Feature Error error respectively. Section C.4 bounds the Bias and combines bounds for the previous terms.

C.1. Sample Error

The bound for this term is summarised within the following Lemma which itself comes from Lemma 1 and 6 in (Carratino et al., 2018).

Lemma 1 (Sample Error) *Under assumptions 2, 4 and 3 , let $\delta \in (0,1)$, $\eta \in (0, \kappa^{-2})$. When*

$$M \geq \big(4 + 18\eta t\big) \log \frac{12\eta t}{\delta}$$

for all $t \geq 1$ with probability atleast $1 - 3\delta$

$$\|S_M\widehat{v}_t - S_M\widetilde{v}_t + S_M\widetilde{v}_t - S_M v_t\|_\rho \leq 4\Big(R\kappa^{2r}\Big(1 + \sqrt{\frac{9}{M}\log\frac{M}{\delta}}\big(\sqrt{\eta t} \vee 1\big)\Big) + \sqrt{B}\Big)$$
$$\times \big(12 + 4\log(t) + \sqrt{2}\eta\big)\Big(\frac{\sqrt{\eta t}}{nm} + \frac{\sqrt{2\sqrt{p}q_0\mathcal{N}(\frac{\kappa^2}{\eta t})}}{\sqrt{nm}}\Big)\log\frac{4}{\delta}$$

where $q_0 = \max\big(2.55, \frac{2\kappa^2}{\|L\|}\big)$

Proof 1 *Apply Lemma 1 in (Carratino et al., 2018) to say $\|S_M\widetilde{v}_t - S_M v_t\|_\rho = 0$, meanwhile Lemma 6 in the same work to bound $\|S_M\widehat{v}_t - S_M\widetilde{v}_t\|$ with $\theta = 0$ and $T = t$.*

C.2. Gradient Descent and Ridge Regression

This term is controlled by Lemma 9 in (Carratino et al., 2018).

Lemma 2 (Gradient Descent and Ridge Regression) *Under Assumption 3 the following holds with probability $1 - \delta$ for $\lambda = \frac{1}{\eta t}$ for $t \geq 1$*

$$\|S_M v_{t+1} - L_M L_{M,\lambda}^{-1} Pf_\rho\|_\rho \leq 8R\kappa^{2r}\Big(\frac{\log\frac{2}{\delta}}{M^r} + \sqrt{\frac{\mathcal{N}\big(\frac{1}{\eta t}\big)^{2r-1}\log\frac{2}{\delta}}{M(\eta t)^{2r-1}}}\Big)\log^{1-r}\big(11\kappa^2\eta t\big) + \frac{2R}{(\eta t)^r}$$

when

$$M \geq (4 + 18\eta t)\log\big(\frac{8\kappa^2\eta t}{\delta}\big)$$

C.3. Random Features Error

The following Lemma is from Lemma 8 of (Rudi & Rosasco, 2017; Carratino et al., 2018).

Lemma 3 *Under assumption 2 and 3 for any $\lambda > 0$, $\delta \in (0, 1/2]$, when*

$$M \geq \left(4 + \frac{18\kappa^2}{\lambda}\right) \log \frac{8\kappa^2}{\lambda\delta}$$

the following holds with probability at least $1 - 2\delta$

$$\|L_M L_{M,\lambda}^{-1} P f_\rho - L L_\lambda^{-1} P f_\rho\|_\rho \leq 4R\kappa^{2r} \left(\frac{\log \frac{2}{\delta}}{M^r} + \sqrt{\frac{\lambda^{2r-1}\mathcal{N}(\lambda)^{2r-1} \log \frac{2}{\delta}}{M}} \right) q^{1-r}$$

where $q = \log \frac{11\kappa^2}{\lambda}$

C.4. Combined Error Bound

The following Lemma combines the error bounds.

Lemma 4 *Under assumption 1 to 4, let $\delta \in (0, 1)$ and $\eta \in (0, \kappa^{-2})$ when*

$$M \geq (4 + 18\eta t\kappa^2) \log \frac{60\kappa^2 \eta t}{\delta}$$

the following holds with probability greater than $1 - \delta$

$$\begin{aligned} \|S_M \widehat{v}_{t+1} - P f_\rho\|_\rho^2 \leq c_1^2 \Big(1 \vee \frac{(\eta t \vee 1) \log \frac{3M}{\delta}}{M}\Big) \Big(\frac{\eta t}{(nm)^2} \vee \frac{\mathcal{N}(\frac{1}{\eta t})}{nm}\Big) \log^2(t) \log^2 \frac{12}{\delta} \\ + c_2^2 \Big(\frac{1}{M^{2r}} \vee \frac{\mathcal{N}(\frac{1}{\eta t})^{2r-1}}{M(\eta t)^{2r-1}}\Big) \log^{2(1-r)}(11\kappa^2 \eta t) \log^2\big(\frac{6}{\delta}\big) + \frac{c_3^2}{(\eta t)^{2r}} \end{aligned}$$

where the constants

$$\begin{aligned} c_1 &= 8 \times 12 \times 15\big(\sqrt{B} \vee (R\kappa^{2r})\big)(1 \vee \sqrt{2\sqrt{p} q_0}) \\ c_2 &= 24R\kappa^{2r} \\ c_3 &= 3R \end{aligned}$$

Proof 2 (Lemma 1) *Begin fixing $\lambda = \frac{1}{\eta t}$ and bounding the bias from Lemma 5 of (Rudi & Rosasco, 2017) as*

$$\|L L_\lambda^{-1} P f_\rho - P f_\rho\|_\rho \leq R\lambda^r.$$

Now use Lemma 1 to bound the Sample Error, Lemma 2 for the Gradient Descent and Ridge Regression Term, and 3 for the Random Features Error. With a union bound, note that the conditions on M for each of these Lemmas is satisfied by $M \geq (4 + 18\eta t\kappa^2) \log \frac{60\kappa^2 \eta t}{\delta}$. Cleaning up constants and squaring then yields the bound.

D. Network Error

In this section we the proof of the following bound on the network error, which improves upon (Richards & Rebeschini, 2019). This section is then structured as follows. Section D.1 provides the error decomposition for the Network Error. Section D.2 introduces a number of prelimary lemmas utilised within the analysis. Section D.3, D.4, D.5, D.6 and D.7 then provides bounds for each of the error terms that arise within the decomposition.

D.1. Error Decomposition

Recall the vector of observations associated to agent $v \in V$ is denoted $\widehat{y}_v = \frac{1}{\sqrt{m}}(y_{1,v}, \dots, y_{m,v})$. Using the previously introduced notation note that we can write the Distributed Gradient Descent iterates as for $t \geq 1$ and $v \in V$

$$\widehat{\omega}_{t+1,v} = \sum_{w \in V} P_{vw}\Big(\widehat{\omega}_{t,w} - \eta \widehat{C}_M^{(w)} \widehat{\omega}_{t,w} + \eta \widehat{S}_M^{(w)\top} \widehat{y}_w\Big)$$

Centering the iterates around the population sequence $\widetilde{v}_t$ we have from the doubly stochastic property of P

$$\begin{aligned}
\widehat{\omega}_{t+1,v} - \widetilde{v}_{t+1} &= \sum_{w \in V} P_{vw}\Big(\widehat{\omega}_{t,w} - \widetilde{v}_t + \eta\big\{(C_M \widetilde{v}_t - S_M^\star f_\rho) - (\widehat{C}_M^{(w)} \widehat{\omega}_{t,w} + \widehat{S}_M^{(w)\top} \widehat{y}_w)\big\}\Big) \\
&= \sum_{w \in V} P_{vw}\Big((I - \widehat{C}_M^{(w)})(\widehat{\omega}_{t,w} - \widetilde{v}_t) + \eta \underbrace{\big\{(C_M \widetilde{v}_t - S_M^\star f_\rho) - (\widehat{C}_M^{(w)} \widetilde{v}_t + \widehat{S}_M^{(w)\top} \widehat{y}_w)\big\}}_{N_{t,w}}\Big) \\
&= \sum_{w \in V} P_{vw}\Big((I - \widehat{C}_M^{(w)})(\widehat{\omega}_{t,w} - \widetilde{v}_t) + \eta N_{t,w}\Big)
\end{aligned}$$

where we have defined the error term

$$N_{t,w} := (C_M \widetilde{v}_t - S_M^\star f_\rho) - (\widehat{C}_M^{(w)} \widetilde{v}_t + \widehat{S}_M^{(w)\star} \widehat{y}_w) \quad \forall s \geq 1 \, w \in V.$$

Note that a similar set of calculation can be performed for the iterates $\widehat{v}_t$ leading to the recursion for $v \in V$ initialised at $\widehat{v}_{1,v} = 0$ and updated for $t \geq 1$

$$\widehat{v}_{t+1,v} - \widetilde{v}_{t+1} = \sum_{w \in V} \frac{1}{n}\Big((I - \widehat{C}_M^{(w)})(\widehat{v}_{t,w} - \widetilde{v}_t) + \eta N_{t,w}\Big)$$

For a path indexed from time step t to k such that $1 \leq k \leq t$ as $w_{t:k} = (w_t, w_{t-1}, \dots, w_k) \in V^{t-k+1}$, let the product of operators be denoted

$$\Pi(w_{t:k}) = (I - \widehat{C}_M^{(w_t)})(I - \widehat{C}_M^{(w_{t-1})}) \dots (I - \widehat{C}_M^{(w_k)}) \tag{10}$$

Meanwhile for $k > t$ we say $\Pi(w_{t:k}) = I$. Unravelling the sequences $\widehat{\omega}_{t+1,v} - \widetilde{v}_{t+1}$ and $\widehat{v}_{t+1} - \widetilde{v}_{t+1}$ with the above notation and taking the difference we then have

$$\begin{aligned}
\widehat{\omega}_{t+1,v} - \widehat{v}_{t+1} &= \sum_{k=1}^{t} \eta \sum_{w_{t:k} \in V^{t-k+1}} \big(P_{v w_{t:k}} - \frac{1}{n^{t-k+1}}\big) \Pi(w_{t:k+1}) N_{k,w_k} \\
&= \sum_{k=1}^{t} \eta \sum_{w_{t:k} \in V^{t-k+1}} \Delta(w_{t:k}) \Pi(w_{t:k+1}) N_{k,w_k}
\end{aligned}$$

where we have introduced the notation where we have denoted $\big(P_{v w_{t:k}} - \frac{1}{n^{t-k+1}}\big) = \Delta(w_{t:k}) \in \mathbb{R}$. Introduce notation for the difference between the product of operators indexed by the paths and the population equivalent

$$\Pi^\Delta(w_{t:k+1}) := \Pi(w_{t:k+1}) - (I - \eta C_M)^{t-k}.$$

Fixing some $t^\star \in \mathbb{N}$ and supposing that $t > 2t^\star \geq 2$, observe that we can then write, for $k \leq t - t^\star - 1$,

$$\begin{aligned}
&\Pi^\Delta(w_{t:k+1}) \\
&= \Pi(w_{t:k+1}) - \Pi(w_{t:k+t^\star+1})(I - \eta C_M)^{t^\star} + \Pi(w_{t:k+t^\star+1})(I - \eta C_M)^{t^\star} - (I - \eta C_M)^{t-k} \\
&= \Pi(w_{t:k+t^\star+1}) \Pi^\Delta(w_{k+t^\star:k+1}) + \Pi^\Delta(w_{t:k+t^\star+1})(I - \eta C_M)^{t^\star}
\end{aligned}$$

where we have replaced the first $t^\star$ operators in $\Pi(w_{t:k})$ with the population variant $(I - \eta C_M)$. Plugging this in then yields

$$\begin{aligned}
\widehat{\omega}_{t+1,v} - \widehat{v}_{t+1} = &\sum_{k=1}^{t} \eta \sum_{w_{t:k} \in V^{t-k+1}} \Delta(w_{t:k})(I - \eta C_M)^{t-k} N_{k,w_k} \\
&+ \sum_{k=t-2t^\star}^{t} \eta \sum_{w_{t:k} \in V^{t-k+1}} \Delta(w_{t:k}) \Pi^\Delta(w_{t:k+1}) N_{k,w_k} \\
&+ \sum_{k=1}^{t-2t^\star-1} \eta \sum_{w_{t:k} \in V^{t-k+1}} \Delta(w_{t:k}) \Pi(w_{t:k+t^\star+1}) \Pi^\Delta(w_{k+t^\star:k+1}) N_{k,w_k} \\
&+ \sum_{k=1}^{t-2t^\star-1} \eta \sum_{w_{t:k} \in V^{t-k+1}} \Delta(w_{t:k}) \Pi^\Delta(w_{t:k+t^\star+1}) (I - \eta C_M)^{t^\star} N_{k,w_k}
\end{aligned}$$

where we split the series off for paths shorter than $2t^\star$. Note for the first and last term above, elements in the series can be simplified by summing over the nodes in the path. Defining for $s \geq 1$ and $v, w \in V$ the difference $\Delta^s(v,w) = P^s_{vw} - \frac{1}{n}$, we get for the first term when $k < t$

$$\begin{aligned}
\sum_{w_{t:k} \in V^{t-k+1}} \Delta(w_{t:k})(I - \eta C_M)^{t-k} N_{k,w_k} &= \sum_{w_k \in V} \Big(\sum_{w_{t:k+1} \in V^{t-k}} \Delta(w_{t:k}) \Big) (I - \eta C_M)^{t-k} N_{k,w_k} \\
&= \sum_{w \in V} \Delta^{t-k}(v,w)(I - \eta C_M)^{t-k} N_{k,w}
\end{aligned}$$

where $\sum_{w_{t:k+1} \in V^{t-k}} \Delta(w_{t:k}) = \sum_{w_{t:k+1} \in V^{t-k}} P_{v w_{t:k}} - \sum_{w_{t:k+1} \in V^{t-k}} \frac{1}{n^{t-k+1}} = P^{t-k}_{vw} - \frac{1}{n} = \Delta^{t-k}(v,w)$. Meanwhile for the last term we can sum over the last $t^\star$ nodes in the path $w_{t:k}$, that is with

$$\begin{aligned}
\sum_{w_{k+t^\star:k+1} \in V^{t^\star}} \Delta(w_{t:k}) &= \sum_{w_{k+t^\star:k+1} \in V^{t^\star}} P_{v w_{t:k}} - \frac{1}{n^{t-k+1}} \\
&= P_{v w_{t:k+t^\star+1}} \sum_{w_{k+t^\star:k+1} \in V^{t^\star}} P_{w_{k+t^\star+1:k}} - \sum_{w_{k+t^\star:k+1} \in V^{t^\star}} \frac{1}{n^{t-k+1}} \\
&= P_{v w_{t:k+t^\star+1}} (P^{t^\star})_{w_{k+t^\star+1} w_k} - \frac{1}{n^{t-t^\star-k+1}} \\
&= P_{v w_{t:k+t^\star+1}} ((P^{t^\star})_{w_{k+t^\star+1} w_k} - \frac{1}{n}) + \frac{1}{n}(P_{v w_{t:k+t^\star+1}} - \frac{1}{n^{t-k-t^\star}}) \\
&= P_{v w_{t:k+t^\star+1}} \Delta^{t^\star}(w_{k+t^\star+1}, w_k) + \frac{1}{n} \Delta(w_{t:k+t^\star+1})
\end{aligned}$$

Plugging this in we get for $1 \leq k \leq t - 2t^\star - 1$

$$\begin{aligned}
&\sum_{w_{t:k} \in V^{t-k+1}} \Delta(w_{t:k}) \Pi^\Delta(w_{t:k+t^\star+1}) (I - \eta C_M)^{t^\star} N_{k,w_k} \\
&= \sum_{w_k \in V} \sum_{w_{t:k+t^\star+1} \in V^{t-t^\star-k}} \Big(\sum_{w_{k+t^\star:k+1} \in V^{t^\star}} \Delta(w_{t:k}) \Big) \Pi^\Delta(w_{t:k+t^\star+1}) (I - \eta C_M)^{t^\star} N_{k,w_k} \\
&= \sum_{w_k \in V} \sum_{w_{t:k+t^\star+1} \in V^{t-t^\star-k}} P_{v w_{t:k+t^\star+1}} \Delta^{t^\star}(w_{k+t^\star+1}, w_k) \Pi^\Delta(w_{t:k+t^\star+1}) (I - \eta C_M)^{t^\star} N_{k,w_k} \\
&\quad + \frac{1}{n} \sum_{w_k \in V} \sum_{w_{t:k+t^\star+1} \in V^{t-t^\star-k}} \Delta(w_{t:k+t^\star+1}) \Pi^\Delta(w_{t:k+t^\star+1}) (I - \eta C_M)^{t^\star} N_{k,w_k} \\
&= \sum_{w_k \in V} \sum_{w_{t:k+t^\star+1} \in V^{t-t^\star-k}} P_{v w_{t:k+t^\star+1}} \Delta^{t^\star}(w_{k+t^\star+1}, w_k) \Pi^\Delta(w_{t:k+t^\star+1}) (I - \eta C_M)^{t^\star} N_{k,w_k} \\
&\quad + \sum_{w_{t:k+t^\star+1} \in V^{t-t^\star-k}} \Delta(w_{t:k+t^\star+1}) \Pi^\Delta(w_{t:k+t^\star+1}) (I - \eta C_M)^{t^\star} N_k
\end{aligned}$$

where at the end for the second term we have

$$\frac{1}{n}\sum_{w_k\in v} N_{k,w_k} = N_k = (C_M\widetilde{v}_t - \mathcal{S}_M^\star f_\rho) - (\widehat{C}_M\widetilde{v}_t + \widehat{S}_M^\top \widehat{y}) \quad \forall k \geq 1.$$

Plugging the above in, using the isometry property (7) and triangle inequality we get

$$\begin{aligned}
\|S_M(\widehat{\omega}_{t+1,v} - \widehat{v}_{t+1})\|_\rho &\leq \sum_{k=1}^{t}\eta\sum_{w\in V}|\Delta^{t-k}(v,w)|\|C_M^{1/2}(I-\eta C_M)^{t-k}N_{k,w}\| \\
&+ \sum_{k=t-2t^\star}^{t}\eta\sum_{w_{t:k}\in V^{t-k+1}}|\Delta(w_{t:k})|\|C_M^{1/2}\Pi^\Delta(w_{t:k+1})N_{k,w_k}\| \\
&+ \sum_{k=1}^{t-2t^\star-1}\eta\sum_{w_{t:k}\in V^{t-k+1}}|\Delta(w_{t:k})|\|C_M^{1/2}\Pi(w_{t:k+t^\star+1})\Pi^\Delta(w_{k+t^\star:k+1})N_{k,w_k}\| \\
&+ \sum_{k=1}^{t-2t^\star-1}\eta\sum_{w_k\in V}\sum_{w_{t:k+t^\star+1}\in V^{t-t^\star-k}}|P_{vw_{t:k+t^\star+1}}\Delta^{t^\star}(w_{k+t^\star+1},w_k)| \\
&\qquad\qquad \times\|C_M^{1/2}\Pi^\Delta(w_{t:k+t^\star+1})(I-\eta C_M)^{t^\star}N_{k,w_k}\| \\
&+ \sum_{k=1}^{t-2t^\star-1}\eta\Big\|\sum_{w_{t:k+t^\star+1}\in V^{t-t^\star-k}}\Delta(w_{t:k+t^\star+1})C_M^{1/2}\Pi^\Delta(w_{t:k+t^\star+1})(I-\eta C_M)^{t^\star}N_k\Big\| \\
&= \mathbf{E_1}+\mathbf{E_2}+\mathbf{E_3}+\mathbf{E_4}+\mathbf{E_5}
\end{aligned} \tag{11}$$

where we have respectively labelled the error terms $\mathbf{E_i}$ for $i = 1,\dots,5$. We will aim to construct high probability bounds for each of these error terms within the following sections. This will rely on utilising the mixing properties of P to control the deviations $\Delta^s(v,w)$ for some $s \geq 1$ and $v, w \in V$, the contractive property of operators $C_M^{1/2}(I-\eta C_M)^k$ for some $k \in \mathbb{N}_+$ as well as concentration of the error terms $N_{k,w}$ and N_k for $k \geq 1$ and $w \in V$. These are summarised within the following section.

D.2. Preliminary Lemmas

In this section we provide some Lemmas that will be useful for later. We begin with the following that bounds the deviation $\Delta^s(v,w)$ in terms of the second largest eigenvalue in absolute value of P.

Lemma 5 (Spectral Bound) *Let $s \geq 1$, $v \in V$. Then the following holds*

$$\sum_{w\in V}|\Delta^s(v,w)| \leq 2(\sqrt{n}\sigma_2^s \wedge 1)$$

Proof 3 (Lemma 5) *Let $e_v \in \mathbb{R}^n$ denoting the standard basis with a 1 in the place associated to agent v. Observe that we can write the deviation in terms of the ℓ_1 norm $\sum_{w\in V}|\Delta^s(v,w)| = \|e_v^\top P^s - \frac{1}{n}\mathbf{1}\|_1$. We immediately have an upper bound from triangle inequality that $\sum_{w\in V}|\Delta^s(v,w)| \leq \|\|e_v^\top P^s\|_1 + \|\frac{1}{n}\mathbf{1}\|_1 = 2$. Meanwhile, we can also go to the ℓ_2 norm and bound*

$$\|e_v^\top P^s - \frac{1}{n}\mathbf{1}\|_1 \leq \sqrt{n}\|e_v^\top P^s - \frac{1}{n}\mathbf{1}\|_2 \leq \sqrt{n}\sigma_2^s.$$

The bound is arrived at by taking the maximum between the two upper bounds.

The following Lemma bonds the norm of contractions

Lemma 6 (Contraction) *Let $\mathcal{L}$ be a compact, positive operator on a separable Hilbert Space H. Assume that $\eta\|\mathcal{L}\| \leq 1$. For $t \in \mathbb{N}$, $a > 0$ and any non-negative integer $k \leq t-1$ we have*

$$\|(I-\eta\mathcal{L})^{t-k}\mathcal{L}^a\| \leq \left(\frac{1}{\eta(t-k)}\right)^a.$$

Proof 4 (Lemma 6) *The proof in Lemma 15 of (Lin & Rosasco, 2017) considers this result with $a = r$. The proof for more general $a > 0$ follows the same steps.*

The following remark will summarise how the above Lemma is applied to control series of contractions.

Remark 5 (Lemma 6) *Lemma 6 will be applied to control series of the form $\eta \sum_{k=1}^{t} \|(I - \eta\mathcal{L})^{t-k}\mathcal{L}^a\|$ for some $t \geq 3$, most notably with powers $a = 1, 1/2$. In the case $a = 1$ we immediately have the bound*

$$\begin{aligned}\eta \sum_{k=1}^{t} \|(I - \eta\mathcal{L})^{t-k}\mathcal{L}\| &= \eta \sum_{k=1}^{t-1} \|(I - \eta\mathcal{L})^{t-k}\mathcal{L}\| + \eta\|\mathcal{L}\| \\ &\leq \eta \sum_{k=1}^{t-1} \frac{1}{\eta(t-k)} + \eta\|\mathcal{L}\| \\ &\leq 5\log(t)\end{aligned}$$

where we have bounded the series $\sum_{k=1}^{t-1} \frac{1}{t-k} \leq 4\log(t)$ and $\eta\|\mathcal{L}\| \leq 1$. Similarly for $a = 1/2$ we have

$$\begin{aligned}\eta \sum_{k=1}^{t} \|(I - \eta\mathcal{L})^{t-k}\mathcal{L}^{1/2}\| &\leq \eta \sum_{k=1}^{t-1} \frac{1}{\sqrt{\eta(t-k)}} + \eta\|\mathcal{L}^{1/2}\| \\ &\leq 3\sqrt{\eta t} + \sqrt{\eta} \\ &\leq 5\sqrt{\eta t}\end{aligned}$$

where we have bounded the series $\sum_{k=1}^{t-1} \frac{1}{\sqrt{t-k)}} \leq 4\sqrt{t}$, see for instance Lemma 23 in (Richards & Rebeschini, 2019) with $q = 0$, as well as the bound that $\sqrt{\eta}\|L^{1/2}\| \leq 1$.

Now for $\lambda > 0$ define the effective dimension associated the feature map ϕ_M, that is

$$\mathcal{N}_M(\lambda) := \mathrm{Tr}\left((L_M + \lambda I)^{-1} L_M\right).$$

Given this, the following Lemma summarises the concentration results used within our analysis.

Lemma 7 (Concentration of Error) *Let $\delta \in (0, 1]$, $n, m, M \in \mathbb{N}_+$, $\lambda > 0$ and $\eta\kappa^2 \leq 1$. Under assumption 2,3 and 4 we have with probability greater than $1 - \delta$ for $1 \leq k \leq t$*

$$\begin{aligned}\max_{w \in V} \|C_{M,\lambda}^{-1/2}(C_M - \widehat{C}_M^{(w)})\| &\leq 2\kappa\Big(\frac{2\kappa}{m\sqrt{\lambda}} + \sqrt{\frac{\mathcal{N}_M(\lambda)}{m}}\Big)\log\frac{6n}{\delta} \\ \max_{w \in V} \|C_{M,\lambda}^{-1/2}N_{k,w}\| &\leq 2\sqrt{B}\Big(\frac{\kappa}{\sqrt{\lambda}m} + \sqrt{\frac{2\sqrt{p}\mathcal{N}_M(\lambda)}{m}}\Big)\log\frac{6n}{\delta} \\ &\quad + 4\kappa\Big(\frac{2\kappa}{m\sqrt{\lambda}} + \sqrt{\frac{\mathcal{N}_M(\lambda)}{m}}\Big)\Big(1 + \sqrt{\frac{9}{M}\log\frac{3Mn}{\delta}}\big(\sqrt{\eta t\kappa} \vee 1\big)\Big)\log\frac{6n}{\delta}\end{aligned}$$

Meanwhile, under the same assumptions with probability greater than $1 - \delta$ for $k \geq 1$

$$\begin{aligned}\|C_{M,\lambda}^{-1/2}(C_M - \widehat{C}_M)\| &\leq 2\kappa\Big(\frac{2\kappa}{nm\sqrt{\lambda}} + \sqrt{\frac{\mathcal{N}_M(\lambda)}{nm}}\Big)\log\frac{2}{\delta} \\ \|C_{M,\lambda}^{-1/2}N_k\| &\leq 2\sqrt{B}\Big(\frac{\kappa}{\sqrt{\lambda}nm} + \sqrt{\frac{2\sqrt{p}\mathcal{N}_M(\lambda)}{nm}}\Big)\log\frac{6}{\delta} \\ &\quad + 4\kappa\Big(\frac{2\kappa}{nm\sqrt{\lambda}} + \sqrt{\frac{\mathcal{N}_M(\lambda)}{nm}}\Big)\Big(1 + \sqrt{\frac{9}{M}\log\frac{3M}{\delta}}\big(\sqrt{\eta t\kappa} \vee 1\big)\Big)\log\frac{6}{\delta}\end{aligned}$$

The proof for this result is given in Section F.1. Lemma 7 will be used extensively within the following analysis. To save on the burden of notation we define the following two functions for $\lambda > 0$, $K \in \mathbb{N}_+$ and $\delta \in (0, 1]$

$$g(\lambda, K) = 2\kappa\Big(\frac{2\kappa}{K\sqrt{\lambda}} + \sqrt{\frac{\mathcal{N}_M(\lambda)}{K}}\Big)$$
$$f(\lambda, K, \delta) = 2\sqrt{B}\Big(\frac{\kappa}{\sqrt{\lambda}K} + \sqrt{\frac{2\sqrt{p}\mathcal{N}_M(\lambda)}{K}}\Big) + 4\kappa\Big(\frac{2\kappa}{K\sqrt{\lambda}} + \sqrt{\frac{\mathcal{N}_M(\lambda)}{K}}\Big)\Big(1 + \sqrt{\frac{9}{M}\log\frac{3M}{\delta}}\big(\sqrt{\eta t\kappa} \vee 1\big)\Big).$$

Looking to Lemma 7 we note the function g is associated to the high probability bound on the difference between the covariance operators, for instance $C_{M,\lambda}^{-1/2}(C_M - \widehat{C}_M)$, meanwhile f is associated to the bound on the error terms, for instance $C_{M,\lambda}^{-1/2}N_k$.

D.3. Bounding $\mathbf{E_1}$

The bound for $\mathbf{E_1}$ is then summarised within the following Lemma.

Lemma 8 (Bounding $\mathbf{E_1}$) *Let $\delta \in (0, 1]$, $n, m, M \in \mathbb{N}_+$ and $\eta\kappa^2 \leq 1$ and $t \geq 2t^\star \geq 2$ and $\lambda, \lambda' > 0$. Under assumption 2,3 and 4 we have with probability greater than $1 - \delta$*

$$\boldsymbol{E}_1 \leq \Big(\|C_{M,\lambda'}^{1/2}\|\sigma_2^{t^\star} t\kappa^{-1} f(\lambda', m, \delta/(2n)) + 20\log(t^\star)(1 \vee \sqrt{\lambda\eta t^\star}) f(\lambda, m, \delta/(2n))\Big)\log\frac{12n}{\delta}$$

Proof 5 (Lemma 8) *Splitting the series at $1 \leq k \leq t - t^\star$ we have the following*

$$\boldsymbol{E}_1 \leq \Big(\max_{1\leq k\leq t, w\in V}\|N_{k,w}\|\Big)\underbrace{\sum_{k=1}^{t-t^\star}\eta\sum_{w\in V}|\Delta^{t-k}(v,w)|\|C_M^{1/2}(I - \eta C_M)^{t-k}\|}_{\boldsymbol{E}_{11}}$$
$$+ \Big(\max_{1\leq k\leq t, w\in V}\|C_{M,\lambda}^{-1/2}N_{k,w}\|\Big)\underbrace{\sum_{k=t-t^\star+1}^{t}\eta\sum_{w\in V}|\Delta^{t-k}(v,w)|\|C_M^{1/2}(I - \eta C_M)^{t-k}C_{M,\lambda}^{1/2}\|}_{\boldsymbol{E}_{12}}$$

To bound $\boldsymbol{E}_{11}$ utilise the mixing properties of the matrix P through Lemma 5. With $\eta\kappa^2 \leq 1$ ensuring that $\eta\|C_M^{1/2}(I - \eta C_M)^{t-k}\| \leq \eta\|C_M^{1/2}\| \leq \sqrt{\eta} \leq \kappa^{-1}$, we arrive at the bound

$$\boldsymbol{E}_{11} \leq \kappa^{-1}\sum_{k=1}^{t-t^\star}\sigma_2^{t-k} \leq \sigma_2^{t^\star} t\kappa^{-1}.$$

Meanwhile to bound $\boldsymbol{E}_{12}$ utilise the contraction of the gradients, that is Lemma 6 remark with $a = 1/2$ and $\mathcal{L} = C_M$. With $\sum_{w\in V}|\Delta^{t-k}(v,w)| \leq 2$ this allows us to say

$$\boldsymbol{E}_{12} \leq 2\eta\sum_{k=t-t^\star+1}^{t}\|C_M(I - \eta C_M)^{t-k}\| + 2\eta\sqrt{\lambda}\sum_{k=t-t^\star+1}^{t}\|C_M^{1/2}(I - \eta C_M)^{t-k}\|$$
$$\leq 20\log(t^\star)(1 \vee \sqrt{\lambda\eta t^\star}).$$

Bounding $\max_{1\leq k\leq t, w\in V}\|N_{k,w}\| \leq \|C_{M,\lambda'}^{1/2}\|\max_{1\leq k\leq t, w\in V}\|C_{M,\lambda'}^{-1/2}N_{k,w}\|$ and plugging in high probability bounds for both $\max_{1\leq k\leq t, w\in V}\|C_{M,\lambda'}^{-1/2}N_{k,w}\|$ and $\max_{1\leq k\leq t, w\in V}\|C_{M,\lambda}^{-1/2}N_{k,w}\|$ from Lemma 7 yields the result.

D.4. Bounding $\mathbf{E}_2$

The bound for this term utilises the following Lemma to bound operator $\|C_M^{1/2}\Pi^{\Delta}(w_{t:k})\|$. To save on national burden, we define the following random quantity for $\lambda > 0$

$$\Delta_\lambda := \max_{v\in V} \|C_M^{-1/2}(C_M - \widehat{C}_M^{(v)})\|.$$

We begin with the following Lemma which rewrites the norm of $\Pi^{\Delta}(w_{t:1})$ for any path $w_{t:1}$ as a series of contractions.

Lemma 9 *Let $N \in \mathbb{R}^M$ and $w_{t:1} \in V^t$ and $\eta\kappa^2 \leq 1$. Then for $u \in [0, 1/2]$*

$$\|C_M^{1/2-u}\Pi^{\Delta}(w_{t:1})N\| \leq 2\eta\Delta_\lambda \|N\| \sum_{\ell=1}^{t} \|C_M^{1/2-u}(I - \eta C_M)^{t-\ell} C_{M,\lambda}^{1/2}\|$$

Given this Lemma we present the high probability bound for $\mathbf{E}_2$.

Lemma 10 (Bounding $\mathbf{E}_2$) *Let $\delta \in (0, 1]$, $n, m, M \in \mathbb{N}_+$ and $\eta\kappa^2 \leq 1$ and $t \geq 2t^\star \geq 2$ and $\lambda, \lambda' > 0$. Under assumption 2,3 and 4 we have with probability greater than $1 - \delta$*

$$\boldsymbol{E}_2 \leq 40\kappa \|C_{M,\lambda'}^{1/2}\| \eta t^\star \log(t)(1 \vee \sqrt{\lambda\eta t}) \log^2 \frac{12n}{\delta} g(\lambda, m) f(\lambda', m, \delta/(2n))$$

Proof 6 (Lemma 10) *Using Lemma 9 with $u = 0$ we have for any $t \geq k \geq t - 2t^\star$ and $w_{t:k} \in V^{t-k+1}$*

$$\begin{aligned}
\|C_M^{1/2}\Pi^{\Delta}(w_{t:k+1})N_{k,w_k}\| &\leq 2\eta\Delta_\lambda \|N_{k,w_k}\| \sum_{\ell=1}^{t-k} \|C_M^{1/2}(I - \eta C_M)^{t-k-\ell} C_{M,\lambda}^{1/2}\| \\
&\leq 2\eta\Delta_\lambda \|N_{k,w_k}\| \Big(\sum_{\ell=1}^{t-k} \|C_M(I - \eta C_M)^{t-k-\ell}\| + \sqrt{\lambda} \sum_{\ell=1}^{t-k} \|C_M^{1/2}(I - \eta C_M)^{t-k-\ell}\| \Big) \\
&\leq 20\eta\Delta_\lambda \|N_{k,w_k}\| \log(t)(1 \vee \sqrt{\lambda\eta t})
\end{aligned}$$

where we applied Lemma 6 remark 5 to the bound the series of contractions. The case $k = t$ the above quantity is zero. With $\sum_{w_{t:k}\in V^{t-k+1}} |\Delta(w_{t:k})| \leq 2$ this leads to the error term being bounded

$$\boldsymbol{E}_2 \leq 40\Delta_\lambda \log(t)(1 \vee \sqrt{\lambda\eta t})\eta t^\star \Big(\max_{1\leq k\leq t, w\in V} \|N_{k,w}\| \Big).$$

The final bound is arrived at by bounding for $\lambda' > 0$ the error term in the brackets as $\max_{1\leq k\leq t, w\in V} \|N_{k,w}\| \leq \|C_{M,\lambda'}^{1/2}\| \max_{1\leq k\leq t, w\in V} \|C_{M,\lambda'}^{-1/2} N_{k,w}\|$, and plugging in high probability bounds for $\max_{1\leq k\leq t, w\in V} \|C_{M,\lambda'}^{-1/2} N_{k,w}\|$ and Δ_λ from Lemma 7, with a union bound.

D.5. Bounding $\mathbf{E}_3$

The bound for this error term is similar to $\mathbf{E}_2$ and will be presented within the following Lemma.

Lemma 11 (Bounding $\mathbf{E}_3$) *Let $\delta \in (0, 1]$, $n, m, M \in \mathbb{N}_+$ and $\eta\kappa^2 \leq 1$ and $t \geq 2t^\star \geq 2$ and $\lambda, \lambda' > 0$. Under assumption 2,3 and 4 we have with probability greater than $1 - \delta$*

$$\boldsymbol{E}_3 \leq 24 \|C_M^{1/2}\| \|C_{M,\lambda'}^{1/2}\| (\eta t)\sqrt{\eta t^\star}\big(1 \vee \sqrt{\lambda\eta t^\star}\big) \log^2 \frac{12n}{\delta} g(\lambda, m) f(\lambda', m, \delta/(2n))$$

Proof 7 (Lemma 11) *For $1 \leq k \leq t - 2t^\star - 1$ and $w_{t:k} \in V^{t-k+1}$ use Lemma 9 with $u = 1/2$ as well as $\eta\kappa^2 \leq 1$ to*

bound with $\lambda > 0$

$$\begin{aligned}
&\|C_M^{1/2}\Pi(w_{t:k+t^\star+1})\Pi^\Delta(w_{k+t^\star:k+1})N_{k,w_k}\| \\
&\leq \|C_M^{1/2}\|\|\Pi^\Delta(w_{k+t^\star:k+1})N_{k,w_k}\| \\
&\leq 2\eta\|C_M^{1/2}\|\Delta_\lambda\|N_{k,w_k}\|\sum_{\ell=1}^{t^\star}\|(I-\eta C_M)^{t^\star-\ell}C_{M,\lambda}^{1/2}\| \\
&\leq 2\|C_M^{1/2}\|\Delta_\lambda\|N_{k,w_k}\|\Big(\eta\sum_{\ell=1}^{t^\star}\|(I-\eta C_M)^{t^\star-\ell}C_M^{1/2}\| + \sqrt{\lambda}\eta t^\star\Big) \\
&\leq 12\|C_M^{1/2}\|\Delta_\lambda\|N_{k,w_k}\|\sqrt{\eta t^\star}(1\vee\sqrt{\lambda\eta t^\star})
\end{aligned}$$

where we have bounded the series of contractions using Lemma 6 remark 5 once again. With $\sum_{w_{t:k}\in V^{t-k+1}}|\Delta(w_{t:k})| \leq 2$*, plugging in the above yields the bound for* $\boldsymbol{E}_3$

$$\boldsymbol{E}_3 \leq 24\|C_M^{1/2}\|(\eta t)\sqrt{\eta t^\star}\Delta_\lambda(1\vee\sqrt{\lambda\eta t^\star})\Big(\max_{1\leq k\leq t, w\in V}\|N_{k,w}\|\Big).$$

The final bound is arrived at by bounding Δ_λ *and* $\Big(\max_{1\leq k\leq t,w\in V}\|N_{k,w}\|\Big)$ *in an identical manner to Lemma 10 for error term* $\boldsymbol{E}_2$*.*

D.6. Bounding E_4

This term will be controlled through the convergence of $P^{t^\star}$ to the stationary distribution. It is summarised within the following Lemma.

Lemma 12 (Bounding E_4) *Let* $\delta \in (0,1]$*,* $n, m, M \in \mathbb{N}_+$ *and* $\eta\kappa^2 \leq 1$ *and* $t \geq 2t^\star \geq 2$ *and* $\lambda > 0$*. Under assumption 2,3 and 4 we have with probability greater than* $1-\delta$

$$\boldsymbol{E}_4 \leq 4\|C_{M,\lambda}^{1/2}\|\big(\sqrt{n}\sigma_2^{t^\star}\wedge 1\big)(\eta t)\log\frac{6n}{\delta}f(\lambda, m, \delta/n)$$

Proof 8 (Lemma 12) *Begin by bounding for* $t-2t^\star-1 \geq k \geq 1$ *,* $w_k \in V$ *and* $w_{t:k+t^\star+1} \in V^{t-t^\star-k}$ *the following*

$$\|C_M^{1/2}\Pi^\Delta(w_{t:k+t^\star+1})(I-\eta C_M)^{t^\star}N_{k,w_k}\| \leq 2\|C_M^{1/2}\|\|N_{k,w_k}\|.$$

Furthermore, we can bound the summation over paths by the deviation of the form $\sum_{w\in V}|\Delta^{t^\star}(v,w)|$ *and use Lemma 5 thereafter to arrive at*

$$\begin{aligned}
\sum_{w_k\in V}\sum_{w_{t:k+t^\star+1}\in V^{t-t^\star-k}}|P_{vw_{t:k+t^\star+1}}\Delta^{t^\star}(w_{k+t^\star+1},w_k)| &= \sum_{w_{t:k+t^\star+1}\in V^{t-t^\star-k}}|P_{vw_{t:k+t^\star+1}}|\Big(\sum_{w_k\in V}|\Delta^{t^\star}(w_{k+t^\star+1},w_k)|\Big) \\
&\leq \max_{u\in V}\Big(\sum_{w\in V}|\Delta^{t^\star}(u,w)|\Big)\Big(\sum_{w_{t:k+t^\star+1}\in V^{t-t^\star-k}}|P_{vw_{t:k+t^\star+1}}|\Big) \\
&= \max_{u\in V}\Big(\sum_{w\in V}|\Delta^{t^\star}(u,w)|\Big) \\
&\leq 2\big(\sqrt{n}\sigma_2^{t^\star}\wedge 1\big).
\end{aligned}$$

Bringing everything together yields the following bound for $\boldsymbol{E}_4$

$$\boldsymbol{E}_4 \leq 2\big(\sqrt{n}\sigma_2^{t^\star}\wedge 1\big)(\eta t)\Big(\max_{1\leq k\leq t,w\in V}\|N_{k,w}\|\Big) \tag{12}$$

Plugging in high probability bounds for $\max_{1\leq k\leq t,w\in V}\|N_{k,w}\|$ *following Lemma 10 for error term* $\boldsymbol{E}_2$ *then yields the bound.*

D.7. Bounding $\mathbf{E}_5$

The summation over paths in this case is decoupled from the error. This allows for a more sophisticated bound to be applied, which considers the deviation of the iterates from the average. The following Lemma effectively bounds the norm of $\sum_{w_{t:1}\in V^t}\Delta(w_{t:1})\Pi^{\Delta}(w_{t:1})$, which involves a sum over the paths $w_{t:1}$.

Lemma 13 *Let $N\in\mathbb{R}^M$, $w_{t:1}\in V^t$ and $\lambda_i\geq 0$ for $i\in\{1,2,3\}$. Then,*

$$\begin{aligned}\|\sum_{w_{t:1}\in V^t}\Delta(w_{t:1})C_M^{1/2}\Pi^{\Delta}(w_{t:1})N\| &\leq 4\eta\Delta_{\lambda_1}\|N\|\sum_{k=1}^{t}\|C_M^{1/2}(I-\eta\widehat{C}_M)^{t-k}C_{M,\lambda_1}^{1/2}\|(\sigma_2^{t-k+1}\wedge 1)\\ &+8\eta^2\Delta_{\lambda_2}\Delta_{\lambda_3}\|N\|\sum_{k=2}^{t}\sum_{\ell=1}^{k-1}\|C_M^{1/2}(I-\eta\widehat{C}_M)^{t-k}C_{M,\lambda_2}^{1/2}\|\|(I-\eta\widehat{C}_M)^{k-1-\ell}C_{M,\lambda_3}^{1/2}\|(\sigma_2^{k-\ell}\wedge 1)\end{aligned}$$

The bound for this error term is then summarised within the following Lemma.

Lemma 14 (Bounding $\mathbf{E}_5$) *Let $\delta\in(0,1]$, $n,m,M\in\mathbb{N}_+$ and $\eta\kappa^2\leq 1$ and $t\geq 2t^\star\geq 2$ and $\lambda',\lambda_i>0$ for $i=1,\dots,3$. Under assumption 2,3 and 4 and if $\frac{9\kappa^2}{M}\log\frac{M}{\delta}\leq\lambda_i$ for $i=1,2$ then with probability greater than $1-8\delta$*

$$\boldsymbol{E}_5\leq\boldsymbol{E}_{51}+\boldsymbol{E}_{52}$$

where

$$\begin{aligned}\boldsymbol{E}_{51}&\leq 84\|C_M^{1/2}C_{M,\lambda_1}^{1/2}\|\|C_{M,\lambda'}^{1/2}\|\eta t(1\vee\sigma_2^{t^\star}\eta t\vee\lambda_1\eta t^\star)\times g(\lambda_1,m)f(\lambda',nm,\delta)\log(t)\log^2\frac{6n}{\delta}\\ \boldsymbol{E}_{52}&\leq 160\|C_{M,\lambda'}^{1/2}\|\|C_{M,\lambda_3}^{1/2}\|(\eta t)(1\vee\lambda_2\eta t)(\sigma_2^{t^\star}\eta t\vee\eta t^\star)\times g(\lambda_2,m)g(\lambda_3,m)f(\lambda',nm,\delta)\log(t)\log^3\frac{6n}{\delta}\end{aligned}$$

Proof 9 (Lemma 14) *Applying for $1\leq k\leq t-2t^\star-1$ Lemma 13 with $N=(I-\eta C_M)^{t^\star}N_k=N_k'$, and $w_{t:k+t^\star+1}\in V^{t-t^\star-k}$ to elements within the series of $\boldsymbol{E}_5$ we arrive at*

$$\begin{aligned}\boldsymbol{E}_5&\leq 4\sum_{k=1}^{t-2t^\star-1}\eta^2\Delta_{\lambda_1}\|N_k'\|\sum_{\ell=1}^{t-t^\star-k}\|C_M^{1/2}(I-\eta\widehat{C}_M)^{t-t^\star-k-\ell}C_{M,\lambda_1}^{1/2}\|\big(\sigma_2^{t-t^\star-k-\ell+1}\wedge 1\big)\\ &+8\sum_{k=1}^{t-2t^\star-1}\eta^3\Delta_{\lambda_2}\Delta_{\lambda_3}\|N_k'\|\sum_{\ell=2}^{t-t^\star-k}\sum_{j=1}^{\ell-1}\|C_M^{1/2}(I-\eta\widehat{C}_M)^{t-t^\star-k-\ell}C_{M,\lambda_2}^{1/2}\|\\ &\qquad\times\|(I-\eta\widehat{C}_M)^{\ell-j-1}C_{M,\lambda_3}^{1/2}\|(\sigma_2^{\ell-j}\wedge 1)\\ &=\boldsymbol{E}_{51}+\boldsymbol{E}_{52}\end{aligned}$$

where we have labelled the remaining error terms $\boldsymbol{E}_{51},\boldsymbol{E}_{52}$. Each of these terms are now bounded.

To bound the first term $\boldsymbol{E}_{51}$, begin by for $1\leq k\leq t-2t^\star-1$ splitting the series at $1\leq\ell\leq t-2t^\star-k$ to arrive at

$$\begin{aligned}&\eta\sum_{\ell=1}^{t-t^\star-k}\|C_M^{1/2}(I-\eta\widehat{C}_M)^{t-t^\star-k-\ell}C_{M,\lambda_1}^{1/2}\|\big(\sigma_2^{t-t^\star-k-\ell+1}\wedge 1\big)\\ &\leq\|C_M^{1/2}C_{M,\lambda_1}^{1/2}\|\eta\sum_{\ell=1}^{t-2t^\star-k}\big(\sigma_2^{t-t^\star-k-\ell+1}\wedge 1\big)+\eta\sum_{\ell=t-2t^\star-k}^{t-t^\star-k}\|C_M^{1/2}(I-\eta\widehat{C}_M)^{t-t^\star-k-\ell}C_{M,\lambda_1}^{1/2}\|\\ &\leq\|C_M^{1/2}C_{M,\lambda_1}^{1/2}\|\eta\sum_{\ell=1}^{t-2t^\star-k}\big(\sigma_2^{t-t^\star-k-\ell+1}\wedge 1\big)+\eta\|C_M^{1/2}\widehat{C}_{M,\lambda_1}^{-1/2}\|\|\widehat{C}_{M,\lambda_1}^{-1/2}C_{M,\lambda_1}^{1/2}\|\sum_{\ell=t-2t^\star-k}^{t-t^\star-k}\|\widehat{C}_{M,\lambda_1}^{1/2}(I-\eta\widehat{C}_M)^{t-t^\star-k-\ell}\widehat{C}_{M,\lambda_1}^{1/2}\|\\ &\leq\|C_M^{1/2}C_{M,\lambda_1}^{1/2}\|\sigma_2^{t^\star}\eta t+10\|C_M^{1/2}\widehat{C}_{M,\lambda_1}^{-1/2}\|\|\widehat{C}_{M,\lambda_1}^{-1/2}C_{M,\lambda_1}^{1/2}\|\log(t)(1\vee\lambda_1\eta t^\star)\end{aligned}$$

where for the first series used that $\sigma_2^{t-t^\star-k-\ell+1} \leq \sigma_2^{t^\star}$ *from* $\ell \leq t - 2t^\star - k$ *meanwhile for the second series*

$$\eta \sum_{\ell=t-2t^\star-k}^{t-t^\star-k} \|\widehat{C}_{M,\lambda_1}^{1/2}(I-\eta\widehat{C}_M)^{t-t^\star-k-\ell}\widehat{C}_{M,\lambda_1}^{1/2}\| \leq \eta \sum_{\ell=t-2t^\star-k}^{t-t^\star-k} \|\widehat{C}_M(I-\eta\widehat{C}_M)^{t-t^\star-k-\ell}\| + \eta\lambda_1 \sum_{\ell=t-2t^\star-k}^{t-t^\star-k} \|(I-\eta\widehat{C}_M)^{t-t^\star-k-\ell}\|$$
$$\leq 5\log(t) + 5\lambda_1\eta t^\star$$

to which we applied Lemma 6 remark 5 to bound the series of contractions. This leads to the bound for $\boldsymbol{E}_{51}$

$$\boldsymbol{E}_{51} \leq 4\Delta_{\lambda_1}\eta t\Big(\|C_M^{1/2}C_{M,\lambda_1}^{1/2}\|\sigma_2^{t^\star}\eta t + 10\|C_M^{1/2}\widehat{C}_{M,\lambda_1}^{-1/2}\|\|\widehat{C}_{M,\lambda_1}^{-1/2}C_{M,\lambda_1}^{1/2}\|\log(t)(1\vee\lambda_1\eta t^\star)\Big)\big(\max_{1\leq k\leq t}\|N_k'\|\big).$$

Provided $\frac{9\kappa^2}{M}\log\frac{M}{\delta} \leq \lambda_1$ *we have from Lemma 3 in (Carratino et al., 2018) that with probability greater than* $1-\delta$

$$\|C_M^{1/2}\widehat{C}_{M,\lambda_1}^{-1/2}\|\|\widehat{C}_{M,\lambda_1}^{-1/2}C_{M,\lambda_1}^{1/2}\| \leq \|\widehat{C}_{M,\lambda_1}^{-1/2}C_{M,\lambda_1}^{1/2}\|^2 \leq 2.$$

Meanwhile for $\lambda' > 0$*, we can bound* $\max_{1\leq k\leq t}\|N_k'\| \leq \|C_{M,\lambda'}^{1/2}\|\max_{1\leq k\leq t}\|C_{M,\lambda'}^{-1/2}N_k'\| \leq \|C_{M,\lambda'}^{1/2}\|\max_{1\leq k\leq t}\|C_{M,\lambda'}^{-1/2}N_k\|$*. The bound is arrived at by also plugging in high probability bounds for* $\|C_{M,\lambda'}^{-1/2}N_k\|$ *and* Δ_{λ_1} *from Lemma 7.*

Finally to bound $\boldsymbol{E}_{52}$*. Begin by bounding for* $1 \leq k \leq t - 2t^\star - 1$ *as well as* $2 \leq \ell \leq t^\star$ *the series as*

$$\sum_{j=1}^{\ell-1}\|(I-\eta\widehat{C}_M)^{\ell-j}C_{M,\lambda_3}^{1/2}\|(\sigma_2^{\ell-j}\wedge 1) \leq \|C_{M,\lambda_3}^{1/2}\|t^\star.$$

Meanwhile for $t^\star + 1 \leq \ell \leq t - t^\star - k$ *we can split the series as* $1 \leq j \leq \ell - t^\star$

$$\begin{aligned}
&\sum_{j=1}^{\ell-1}\|(I-\eta\widehat{C}_M)^{\ell-j}C_{M,\lambda_3}^{1/2}\|(\sigma_2^{\ell-j}\wedge 1)\\
&\leq \|C_{M,\lambda_3}^{1/2}\|\sum_{j=1}^{\ell-t^\star}(\sigma_2^{\ell-j}\wedge 1) + \sum_{j=\ell-t^\star+1}^{\ell-1}\|(I-\eta\widehat{C}_M)^{\ell-j}C_{M,\lambda_3}^{1/2}\|\\
&\leq \|C_{M,\lambda_3}^{1/2}\|(\sigma_2^{t^\star}t + t^\star)
\end{aligned}$$

where for the first series we applied $j \leq \ell - t^\star$ *to say* $\sigma_2^{\ell-j} \leq \sigma_2^{t^\star}$*, and for the second simply summed up the* $t^\star$ *terms after bounding* $\|(I-\eta\widehat{C}_M)^{\ell-j}C_{M,\lambda_3}^{1/2}\| \leq \|C_{M,\lambda_3}^{1/2}\|$*. Plugging in the above bound for all* $2 \leq \ell \leq t - t^\star - k$ *we arrive at the following bound for* $\boldsymbol{E}_{52}$

$$\boldsymbol{E}_{52} \leq 8\Delta_{\lambda_2}\Delta_{\lambda_3}\big(\max_{1\leq k\leq t}\|N_k'\|\big)\|C_{M,\lambda_3}^{1/2}\|(\sigma_2^{t^\star}\eta t + \eta t^\star)\sum_{k=1}^{t-2t^\star-1}\eta^2\sum_{\ell=2}^{t-t^\star-k}\|C_M^{1/2}(I-\eta\widehat{C}_M)^{t-t^\star-k-\ell}C_{M,\lambda_2}^{1/2}\|$$

For $1 \leq k \leq t - 2t^\star - 1$ *the series of contractions over* ℓ *can be bounded using Lemma 6 remark 5 in a similar manner to previously as*

$$\eta\sum_{\ell=2}^{t-t^\star-k}\|C_M^{1/2}(I-\eta\widehat{C}_M)^{t-t^\star-k-\ell}C_{M,\lambda_2}^{1/2}\| \leq \|C_M^{1/2}\widehat{C}_{M,\lambda_2}^{-1/2}\|\|\widehat{C}_{M,\lambda_2}^{-1/2}C_{M,\lambda_2}^{1/2}\|10\log(t)(1\vee\lambda_2\eta t).$$

Summing up the remaining series for over k*, using that* $\|C_M^{1/2}\widehat{C}_{M,\lambda_2}^{-1/2}\|\|\widehat{C}_{M,\lambda_2}^{-1/2}C_{M,\lambda_2}^{1/2}\| \leq 2$ *from* $\frac{9\kappa^2}{M}\log\frac{M}{\delta} \leq \lambda_2$*, plugging in high probability bounds for* $\max_{1\leq k\leq t}\|N_k'\|$ *from the the error term* $\boldsymbol{E}_{51}$*, as well as high probability bounds for* $\Delta_{\lambda_2}, \Delta_{\lambda_3}$ *from Lemma 7 yields the bound.*

E. Final bounds

In this section we bring together the high probability bounds for the Statistical Error and Distributed Error. This section is then as follows. Section E.1 provides the proof for Theorem 1. Section E.2 gives the proof for Theorem 1.

E.1. Refined Bound (Theorem 2)

In this section we give conditions under which we obtained a refined bound.

Proof 10 (Theorem 2) *Fixing $\delta \in (0,1]$ and a constant $c_{union} > 1$, assume that*

$$\begin{aligned}
&\eta t = (nm)^{\frac{1}{2r+\gamma}} \\
&M \geq \Big((nm)^{\frac{1+\gamma(2r-1)}{2r+\gamma}}\Big) \vee \Big(\eta t \log \frac{60 n \kappa^2 (\eta t \vee M) c_{union}}{\delta}\Big) \\
&t^\star \geq 2\frac{\log(nmt)}{1-\sigma_2} \\
&m \geq \Big((1 \vee (\eta t^\star))^{2r+\gamma} n^{2r/\gamma}\Big) \vee \Big((1 \vee (\eta t^\star))^2 n\Big) \vee \Big((1 \vee \eta t^\star)^{\frac{(1+\gamma)(2r+\gamma)}{2(r+\gamma-1)}} n^{\frac{(r+1)}{(r+\gamma-1)}}\Big)
\end{aligned}$$

Now, consider the error decomposition given (8), to arrive at the bound

$$\mathcal{E}(f_{t+1,v}) - \mathcal{E}(f_{\mathcal{H}}) \leq 2 \underbrace{\|S_M \widehat{\omega}_{t+1,v} - S_M \widehat{v}_t\|_\rho^2}_{(\text{Network Error})^2} + 2 \underbrace{\|S_M \widehat{v}_t - P f_\rho\|_\rho^2}_{(\text{Statistical Error})^2}.$$

Begin by bounding the statistical error by using Lemma 4. Using Assumption 5 to bound $\mathcal{N}(\frac{1}{\eta t}) \leq Q^2(\eta t)^\gamma$, and noting that $M \geq (4 + 18\eta t \kappa^2) \log \frac{60\kappa^2 \eta t}{\delta}$ is satisfied, allows us to upper bound with probability greater than $1-\delta$

$$\begin{aligned}
\|S_M \widehat{v}_t - P f_\rho\|_\rho^2 \leq (nm)^{-2r/(2r+\gamma)} &\Big(c_1^2 \Big(1 \vee \frac{(\eta t) \log \frac{3M}{\delta}}{M}\Big)(1 \vee Q^2) \log^2(t) \log^2\Big(\frac{12}{\delta}\Big) + c_3^2\Big) \\
&+ c_2^2 \Big(\frac{1}{M^{2r}} \vee \frac{Q^2}{M(nm)^{(1-\gamma)(2r-1)/(2r+\gamma)}}\Big) \log^{2(1-r)}(11\kappa^2 \eta t) \log^2\big(\frac{6}{\delta}\big)
\end{aligned}$$

The quantity within the brackets for second term is then upper bounded $\frac{1}{M(nm)^{(1-\gamma)(2r-1)/(2r+\gamma)}} \leq (nm)^{-2r/(2r+\gamma)}$ provided $M \geq (nm)^{\frac{1+\gamma(2r-1)}{2r+\gamma}}$, which is satisfied as an assumption in the Theorem. This results in an upper bound on the statistical error that is, up to log factors, decreasing as $(nm)^{-2r/(2r+\gamma)}$ in high probability.

We now proceed to bound the Network Error Term. Begin by considering error decomposition given in (11) into the terms $\boldsymbol{E}_1, \boldsymbol{E}_2, \boldsymbol{E}_3, \boldsymbol{E}_4, \boldsymbol{E}_5$, in particular by applying the inequality $(a+b)^2 \leq 2a^2 + 2b^2$ multiple times we get

$$\|S_M \widehat{\omega}_{t+1,v} - S_M \widehat{v}_t\|_\rho^2 \leq 2\boldsymbol{E}_1^2 + 4\boldsymbol{E}_2^2 + 8\boldsymbol{E}_3^2 + 16\boldsymbol{E}_4^2 + 32\boldsymbol{E}_5^2,$$

and thus it is sufficient to show each of these terms is decreasing as $(nm)^{-2r/(2r+\gamma)}$ in high probability. Before doing so we note Lemma 4 in (Carratino et al., 2018) states for any $\lambda > 0$ that if

$$M \geq \big(4 + \frac{18\kappa^2}{\lambda}\big) \log \frac{12\kappa^2}{\lambda\delta}$$

then with probability greater than $1-\delta$ we have $\mathcal{N}_M(\lambda) \leq q\mathcal{N}(\lambda)$ where $q = \max\big(2.55, \frac{2\kappa^2}{\|L\|}\big)$. We note this is satisfied with both $\lambda = (\eta t)^{-1}, (1 \vee (\eta t^\star))^{-1}$ by the assumptions within the Theorem, and as such, we can interchange from $\mathcal{N}_M(\lambda)$ to $\mathcal{N}(\lambda)$ with at most a constant cost of q.

We begin by bounding $\boldsymbol{E}_1^2$ by considering Lemma 8 with $\lambda' = \kappa^2$ and $\lambda = (1 \vee \eta t^\star)^{-1}$, which leads to with probability greater than $1-\delta$

$$\boldsymbol{E}_1^2 \leq \Big(2\|C_{M,\lambda'}^{1/2}\|^2 \sigma_2^{2t^\star} t^2 \kappa^{-2} (f(\lambda', m, \delta/(2n)))^2 + 40 \log^2(t^\star)(f(\lambda, m, \delta/(2n)))^2\Big) \log^2 \frac{12n}{\delta}$$

Now due to $t^\star \geq \frac{2\log(nmt)}{1-\sigma_2} \geq \frac{2\log(nmt)}{-\log(\sigma_2)}$ (the second inequality arising from $\log(x) \geq 1 - x^{-1}$ for $x \geq 0$) we have $\sigma_2^{t^\star} \leq (tnm)^{-2}$. As such with the fact that $f(\kappa^2, m, \delta/(2n)) \lesssim m^{-1/2}$ in high probability, the first term above is decreasing, upto logarithmic factors, as $(nm)^{-2r/(2r+\gamma)}$. Meanwhile for the second term we have that

$$f((1 \vee \eta t^\star)^{-1}, m, \delta/2)^2 \leq a_1^2 \big(\frac{(1 \vee \eta t^\star)}{m^2} \vee \frac{(1 \vee \eta t^\star)^\gamma}{m}\big)\big(1 \vee \frac{3(\eta t \kappa \vee 1)}{M} \log \frac{6Mn}{\delta}\big)$$

for the constant $a_1 = 64\Big(\sqrt{B}(\kappa \vee \sqrt{\sqrt{pq}})\Big) \vee \big(\kappa \vee \sqrt{q}\big)$. *For* $\boldsymbol{E}_1^2$ *to be decreasing at the rate* $(nm)^{-2r/(2r+\gamma)}$, *up to logarithmic factors, we then require* $\frac{(1\vee\eta t^\star)^\gamma}{m} \leq (nm)^{-2r/(2r+\gamma)}$ *which is satisfied when* $m \geq (1 \vee (\eta t^\star))^{2r+\gamma} n^{2r/\gamma}$.

Proceed to bound $\boldsymbol{E}_2^2$ *by considering Lemma 10 with* $\lambda = 1/(\eta t)$ *and* $\lambda' = \kappa^2$ *to arrive at with probability greater than* $1-\delta$

$$\boldsymbol{E}_2^2 \leq 40^2\kappa^2\|C_{M,\lambda'}^{1/2}\|^2 \log^2(t)(\eta t^\star)^2 (g(\lambda, m))^2 (f(\lambda', m, \delta/(2n)))^2 \log^4 \frac{12n}{\delta}$$

As discussed previously, we have with high probability that $(f(\kappa^2, m, \delta/(2n)))^2 \lesssim 1/m$, *meanwhile*

$$g((\eta t)^{-1}, m)^2 \leq a_2^2\big(\frac{\eta t}{m^2} \vee \frac{(\eta t)^\gamma}{m}\big)$$

where $a_2 = 8\kappa(\kappa \vee \sqrt{q})$. *As such for* $\boldsymbol{E}_2^2$ *to be decreasing at the rate* $(nm)^{-2r/(2r+\gamma)}$ *we require* $\frac{(\eta t)^\gamma (1\vee\eta t^\star)^2}{m^2} \leq (nm)^{-2r/(2r+\gamma)}$ *which, plugging in* $\eta t = (nm)^{1/(2r+\gamma)}$ *is satisfied when* $m \geq (1 \vee \eta t^\star)^2 n$.

Bounding $\boldsymbol{E}_3$ *using Lemma 11 with* $\lambda = (1 \vee (\eta t^\star))^{-1}$ *and* $\lambda' = \kappa^2$ *we have with probability greater than* $1-\delta$

$$\boldsymbol{E}_3^2 \leq 24^2\|C_M^{1/2}\|^2\|C_{M,\lambda'}^{1/2}\|^2(\eta t)^2(\eta t^\star)(g(\lambda, m))^2(f(\lambda', m, \delta/(2n)))^2 \log^4 \frac{12n}{\delta}.$$

Following the steps for $\boldsymbol{E}_2$, *we have with high probability that* $f(\kappa^2, m, \delta/(2n))^2 \lesssim 1/m$, *meanwhile* $g((1\vee(\eta t^\star))^{-1}, m) \lesssim (1\vee\eta t^\star)^\gamma/m$. *As such for* $\boldsymbol{E}_3^2$ *to be decreasing with the rate* $(nm)^{-2r/(2r+\gamma)}$ *we require* $\frac{(\eta t)^2(1\vee\eta t^\star)^{1+\gamma}}{m^2} \leq (nm)^{-2r/(2r+\gamma)}$, *which is satisfied when* $r+\gamma > 1$ *and* $m \geq (1\vee\eta t^\star)^{\frac{(1+\gamma)(2r+\gamma)}{2(r+\gamma-1)}} n^{\frac{(r+1)}{(r+\gamma-1)}}$.

Now to bound $\boldsymbol{E}_4$ *we consider Lemma 12 with* $\lambda = \kappa^2$ *to arrive at with probability greater than* $1-\delta$

$$\boldsymbol{E}_4^2 \leq 16\|C_{M,\lambda}^{1/2}\|^2(n\sigma_2^{2t^\star} \wedge 1)(\eta t)^2 \log^2\Big(\frac{6n}{\delta}\Big)(f(\lambda, m, \delta/n))^2.$$

Following the previous analysis we know with high probability $(f(\lambda, m, \delta/n))^2 = \widetilde{O}(1/m)$ *and that* $t^\star$ *is such that* $\sigma_2^{t^\star} \leq (tnm)^{-2}$. *Combining these two facts we have that* $\boldsymbol{E}_4^2$ *is of the order* $(nm)^{-2r/(2r+\gamma)}$ *with high probability.*

The bound for $\boldsymbol{E}_5^2$ *is naturally split across the terms* $\boldsymbol{E}_{51}, \boldsymbol{E}_{52}$ *from Lemma 14. In particular we have that*

$$\boldsymbol{E}_5^2 \leq 2\boldsymbol{E}_{51}^2 + 2\boldsymbol{E}_{52}^2$$

The remainder of the proof then shows each of the terms above are decreasing at the rate $(nm)^{-2r/(2r+\gamma)}$ *in high probability by using the bounds provided within Lemma 14. We note the condition* $\frac{9\kappa^2}{M}\log\frac{M9\kappa^2}{\delta} \leq \lambda_i$ *for* $i = 1, 2$ *is satisfied for* $\lambda_1 = (1 \vee (\eta t^\star))^{-1}$ *and* $\lambda_2 = (\eta t)^{-1}$ *by the assumptions.*

Consider the bound for $\boldsymbol{E}_{51}$ *with* $\lambda_1 = (1 \vee (\eta t^\star))^{-1}$ *and* $\lambda' = \kappa^2$, *so we have with probability greater than* $1-\delta$

$$\boldsymbol{E}_{51}^2 \leq 84^2\|C_M^{1/2}C_{M,\lambda_1}^{1/2}\|^2\|C_{M,\lambda'}^{1/2}\|^2(\eta t)^2(1 \vee \sigma_2^{2t^\star}(\eta t)^2)(g(\lambda_1, m))^2(f(\lambda', nm, \delta/8))^2 \log^2(t)\log^4\frac{48n}{\delta}.$$

From previously we have that $t^\star$ *so that* $\sigma_2^{t^\star} \leq (tnm)^{-2}$ *and thus* $\sigma_2^{t^\star}\eta t \leq 1$. *Meanwhile following steps from previously we have* $(g(\lambda_1, m))^2 \lesssim (1 \vee (\eta t^\star))^\gamma/m$ *as well as with high probability* $(f(\lambda', nm, \delta))^2 \lesssim (nm)^{-1}$. *As such we require* $\frac{(\eta t)^2(1\vee(\eta t^\star))^\gamma}{m(nm)} \leq (nm)^{-2r/(2r+\gamma)}$ *which is satisfied when* $r+\gamma > 1$ *and* $m \geq n^{\frac{2-\gamma}{2(r+\gamma-1)}}(1 \vee (\eta t^\star))^{\frac{\gamma(2r+\gamma)}{2(r+\gamma-1)}}$. *This is then implied by the assumption that* $m \geq (1\vee\eta t^\star)^{\frac{(1+\gamma)(2r+\gamma)}{2(r+\gamma-1)}} n^{\frac{(r+1)}{(r+\gamma-1)}}$ *and* $r+\gamma \geq 1$.

Finally to bound $\boldsymbol{E}_{52}$ *consider the bound given with* $\lambda_2 = (\eta t)^{-1}$, *and* $\lambda_3 = \lambda' = \kappa^2$ *to arrive at with probability greater than* $1-\delta$

$$\boldsymbol{E}_{52}^2 \leq 160^2\|C_{M,\lambda'}^{1/2}\|^2\|C_{M,\lambda_3}^{1/2}\|^2(\eta t)^2\big(\sigma_2^{t^\star}\eta t \vee (\eta t^\star)^2\big)g(\lambda_2, m)g(\lambda_3, m)f(\lambda', nm, \delta/8)\log^2(t)\log^6\frac{48n}{\delta}.$$

Once again $\sigma_2^{t^\star} \leq (tnm)^{-2}$ *ensures* $\sigma_2^{t^\star}\eta t \leq (1\vee\eta t^\star)$. *Meanwhile we have* $(g(\lambda_2, m))^2 \lesssim (\eta t)^\gamma/m$, $(g(\lambda_3, m))^2 \lesssim 1/m$ *and with high probability* $(f(\lambda', nm, \delta/8))^2 \lesssim 1/(nm)$. *As such to ensure this term is sufficiently small we require*

$\frac{(\eta t)^{2+\gamma}(1\vee\eta t^\star)^2}{m^2(nm)} \leq (nm)^{-2r/(2r+\gamma)}$, *which satisfied if* $m \geq n^{\frac{1}{2r+\gamma}}(1\vee(\eta t^\star))^{\frac{2r+\gamma}{2r+\gamma-1}}$. *This then being implied by* $m \geq (1\vee\eta t^\star)^{\frac{(1+\gamma)(2r+\gamma)}{2(r+\gamma-1)}} n^{\frac{(r+1)}{(r+\gamma-1)}}$ *since* $\frac{r+1}{r+\gamma-1} \geq \frac{1}{2r+\gamma}$ *and* $\frac{(1+\gamma)(2r+\gamma)}{2(r+\gamma-1)} \geq \frac{2r+\gamma}{2r+\gamma-1}$. *The second inequality arising from the observation that* $\frac{1}{2(r+\gamma-1)} \geq \frac{1}{2(r+\gamma-1)+1-\gamma} = \frac{1}{2r+\gamma-1}$.

Each of the bounds for $\boldsymbol{E}_i^2$ *for* $i = 1,\dots,5$ *hold in high probability, and as such, can be combined with a union bound. This incurs at most a logarithmic factor in the bound, with the number of unions applied being upper bounded by the constant* $c_{union} > 1$ *chosen at the start.*

E.2. Worst Case (Theorem 1)

Consider the refined bound in Theorem 2 with $r = 1/2$ and $\gamma = 1$.

E.3. Leading Order Error Terms (Theorem 3)

Follow the proof of Theorem 2, where the error is decomposed into the following terms

$$\mathcal{E}(f_{t+1,v}) - \mathcal{E}(f_{\mathcal{H}}) \leq (\text{Network Error})^2 + (\text{Statistical Error})^2.$$

The statistical error follows (Carratino et al., 2018) and, in our work, is summarised within Lemma 4 to be upto logarithmic factors in high-probability

$$(\text{Statistical Error})^2 \lesssim \underbrace{(1\vee\frac{\eta t}{M})\frac{(\eta t)^\gamma}{nm}}_{\text{Sample Variance}} + \underbrace{\frac{1}{M(\eta t)^{(1-\gamma)(2r-1)}}}_{\text{Random Fourier Error}} + \underbrace{\frac{1}{(\eta t)^{2r}}}_{\text{Bias}}.$$

Meanwhile the network error is bounded into terms

$$(\text{Network Error})^2 \lesssim \mathbf{E}_1^2 + \mathbf{E}_2^2 + \mathbf{E}_3^2 + \mathbf{E}_4^2 + \mathbf{E}_5^2$$

where high-probability bounds from Section D are used. In particular, the bounds each term are, up to logarithmic factors, in high probability

$$\begin{aligned}
\mathbf{E}_1^2 &\lesssim \frac{(\eta t^\star)^\gamma}{m} \\
\mathbf{E}_2^2 &\lesssim \frac{(\eta t^\star)^2(\eta t)^\gamma}{m^2} \\
\mathbf{E}_3^2 &\lesssim \frac{(\eta t)^2(\eta t^\star)^{1+\gamma}}{m^2} \\
\mathbf{E}_4^2 &\lesssim \frac{n\sigma_2^{2t^\star}(\eta t)^2}{m} \\
\mathbf{E}_5^2 &\lesssim \frac{(\eta t)^2(1\vee(\eta t^\star))^\gamma}{m(nm)} + \frac{(\eta t)^{2+\gamma}(1\vee\eta t^\star)^2}{m^2(nm)}
\end{aligned}$$

The leading order terms are then defined as $\mathbf{E}_1^2$ and $\mathbf{E}_3^2$.

F. Proofs of Auxiliary Lemmas

In this section we provide the proofs of the auxiliary lemmas. This section is then as follows. Section F.1 provides the proof for Lemma 7. Section F.2 provides the proof of Lemma 9. Section F.3 provides the proof of Lemma 13.

F.1. Concentration of Error terms (Lemma 7)

Proof 11 (Lemma 7) *Fix* $w \in V$. *We begin by collecting the necessary concentration results. Following Lemma 18 in (Lin & Cevher, 2018) with* $\mathcal{T}_\rho, \mathcal{T}_{\mathbf{x}}$ *swapped for* $C_M, \widehat{C}_M^{(w)}$ *respectively (or Proposition 5 in (Rudi & Rosasco, 2017)) we have with probability greater than* $1-\delta$

$$\|C_{M,\lambda}^{-1/2}(C_M - \widehat{C}_M^{(w)})\| \leq 2\kappa\Big(\frac{2\kappa}{m\sqrt{\lambda}} + \sqrt{\frac{\mathcal{N}_M(\lambda)}{m}}\Big)\log\frac{2}{\delta}$$

From Lemma 2 in (Carratino et al., 2018) under assumptions 2 and 3 we have with probability greater than $1-\delta$ *for all* $t \geq 1$

$$\|\widetilde{v}_{t+1}\| \leq 2R\kappa^{2r-1}\Big(1+\sqrt{\frac{9\kappa^2}{M}\log\frac{M}{\delta}}\max\Big(\sqrt{\eta t},\kappa^{-1}\Big)\Big).$$

Meanwhile from Lemma 6 in (Rudi & Rosasco, 2017) under assumption 2 and 4 we have with probability greater than $1-\delta$

$$\|C_{M,\lambda}^{-1/2}(\widehat{S}_M^{(w)\top}\widehat{y}-S_M^\star f_\rho)\| \leq 2\sqrt{B}\Big(\frac{\kappa}{\sqrt{\lambda}m}+\sqrt{\frac{2\sqrt{p}\mathcal{N}_M(\lambda)}{m}}\Big)\log\frac{2}{\delta}$$

Considering $\|C_{M,\lambda}^{-1/2}N_{k,w}\|$*, using triangle inequality and plugging the above bounds with a union bound, we have with probability greater than* $1-\delta$

$$\begin{aligned}\|C_{M,\lambda}^{-1/2}N_{k,w}\| &\leq \|C_{M,\lambda}^{-1/2}(C_M-\widehat{C}_M^{(w)})\|\|\widetilde{v}_{t+1}\|+\|C_{M,\lambda}^{-1/2}(\widehat{S}_M^{(w)\star}\widehat{y}-S_M^\star f_\rho)\| \\ &\leq 2\kappa\Big(\frac{2\kappa}{m\sqrt{\lambda}}+\sqrt{\frac{\mathcal{N}_M(\lambda)}{m}}\Big)\log\frac{6}{\delta}\Big(1+\sqrt{\frac{9\kappa^2}{M}\log\frac{3M}{\delta}}\max\Big(\sqrt{\eta t},\kappa^{-1}\Big)\Big) \\ &\quad+2\sqrt{B}\Big(\frac{\kappa}{\sqrt{\lambda}m}+\sqrt{\frac{2\sqrt{p}\mathcal{N}_M(\lambda)}{m}}\Big)\log\frac{6}{\delta}.\end{aligned}$$

Now a bound over the maximum $\max_{w\in V}\|C_{M,\lambda}^{-1/2}N_{k,w}\|$ *is obtained by taking a union bound over* $w \in V$*. Meanwhile, an identical set of steps with* $\widehat{C}_M^{(w)}, \widehat{S}_M^{(w),\top}$ *swapped for* $\widehat{C}_M, \widehat{S}_M$ *yields the bound for* $\|C_{M,\lambda}^{-1/2}N_k\|$ *and* $\|C_{M,\lambda}^{-1/2}(C_M-\widehat{C}_M)\|$*.*

F.2. Difference between Product of Empirical and Population Operators (Lemma 9)

In this section we provide the proof for Lemma 9.

Proof 12 (Lemma 9) *Begin by writing the quantity* $\Pi^\Delta(w_{t:1})N$ *using two auxiliary sequences. Initialized at* $\gamma_1=\gamma_1'=N$ *and updated for* $t\geq s\geq 1$ *we have*

$$\begin{aligned}\gamma_{s+1}' &= (I-\eta\widehat{C}_M^{(w_s)})\gamma_s' = \Pi(w_{s:1})N \\ \gamma_{s+1} &= (I-\eta C_M)\gamma_s = (I-\eta C_M)^s N\end{aligned}$$

We can then write the difference as between these two sequences as the recursion

$$\begin{aligned}\gamma_{s+1}'-\gamma_{s+1} &= (I-\eta C_M)(\gamma_s'-\gamma_s)+\eta\{C_M-\widehat{C}_M^{(w_s)}\}\gamma_s' \\ &= (I-\eta C_M)^s(\gamma_1'-\gamma_1)+\sum_{\ell=1}^{s}\eta(I-\eta C_M)^{s-\ell}\{C_M-\widehat{C}_M^{(w_\ell)}\}\gamma_\ell' \\ &= \sum_{\ell=1}^{s}\eta(I-\eta C_M)^{s-\ell}\{C_M-\widehat{C}_M^{(w_\ell)}\}\gamma_\ell'.\end{aligned}$$

We then have

$$\begin{aligned}\|C_M^{1/2-u}\Pi^\Delta(w_{t:1})N\| &= \|C_M^{1/2-u}(\gamma_{t+1}'-\gamma_{t+1})\| \\ &= \|\sum_{\ell=1}^{t}\eta C_M^{1/2-u}(I-\eta C_M)^{t-\ell}\{C_M-\widehat{C}_M^{(w_\ell)}\}\gamma_\ell'\| \\ &\leq \sum_{\ell=1}^{t}\eta\|C_M^{1/2-u}(I-\eta C_M)^{t-\ell}C_{M,\lambda}^{1/2}\|\|C_{M,\lambda}^{-1/2}(C_M-\widehat{C}_M^{(w_\ell)})\|\|\gamma_\ell'\| \\ &\leq \Delta_\lambda\|N\|\sum_{\ell=1}^{t}\eta\|C_M^{1/2-u}(I-\eta C_M)^{t-\ell}C_{M,\lambda}^{1/2}\|\end{aligned}$$

where we have taken out the maximum over the $w_\ell \in V$ *for* $\|C_{M,\lambda}^{-1/2}(C_{M,\lambda}-\widehat{C}_M^{(w_\ell)})\|$ *and simply bounded* $\|\gamma_\ell'\| = \|(I-\eta\widehat{C}_M^{(w_{\ell-1})})\gamma_{\ell-1}'\| \leq \|\gamma_{\ell-1}'\| \leq \|N\|$ *from* $\eta\kappa^2\leq 1$*.*

F.3. Convolution of Difference between Product of Empirical and Population Operators (Lemma 13)

This section provides the proof of Lemma 13.

Proof 13 (Lemma 13) *Begin by observing that this quantity can be written as*

$$\sum_{w_{t:1}\in V^t}\Delta(w_{t:1})\Pi^{\Delta}(w_{t:1})N = \sum_{w_{t:1}\in V^t}\Delta(w_{t:1})\Pi(w_{t:1})N - \sum_{w_{t:1}\in V^t}\Delta(w_{t:1})(I-\eta C_M)^t N$$
$$= \sum_{w_{t:1}\in V^t}\Delta(w_{t:1})\Pi(w_{t:1})N$$

since $\sum_{w_{t:1}\in V^t}\Delta(w_{t:1})=0$. *Now introduce the following auxiliary variables. Initialized as* $\gamma_{1,w}=\gamma'_{1,w}=N$ *for all* $w\in V$ *we update the sequences for* $t\geq s\geq 1$

$$\gamma_{s+1,v} = \sum_{w\in V}P_{vw}(I-\eta\widehat{C}_M^{(w)})\gamma_{s,w} = \sum_{w_{s:1}\in V^s}P_{vw_{s:1}}\Pi(w_{s:1})N \tag{13}$$
$$\gamma'_{s+1,v} = \sum_{w\in V}\frac{1}{n}(I-\eta\widehat{C}_M^{(w)})\gamma'_{s,w} = \sum_{w_{s:1}\in V^s}\frac{1}{n^s}\Pi(w_{s:1})N.$$

The quantity bounded within Lemma 13 can then be seen as the difference

$$\|C_M^{1/2}(\gamma_{t+1,v}-\gamma'_{t+1,v})\| = \Big\|\sum_{w_{t:1}\in V^t}\Delta(w_{t:1})C_M^{1/2}\Pi(w_{t:1})N\Big\|.$$

Introducing the auxiliary sequence $\{\gamma'_s\}_{s\geq 1}$ *independent of the agents. Also initialised* $\gamma'_{1,w}=N=:\gamma'_1$ *for all* $w\in V$ *we have due to averaging over all of the agents uniformly* $\gamma'_{2,w}=\gamma'_2=(I-\eta\widehat{C}_M)N$ *for all* $w\in V$. *Applying this recursively we have for* $s\geq 1$ *and* $v\in V$

$$\gamma'_{s+1,v}=\gamma'_{s+1}=(I-\eta\widehat{C}_M)^sN.$$

Combined with the fact that the iterates $\{\gamma_{s,v}\}_{s\in[t],v\in V}$ *can be written and unravelled*

$$\gamma_{t+1,v} = \sum_{w\in V}P_{vw}\big((I-\eta\widehat{C}_M)\gamma_{t,w}+\eta\{\widehat{C}_M-\widehat{C}_M^{(w)}\}\gamma_{t,w}\big)$$
$$= (I-\eta\widehat{C}_M)^tN+\eta\sum_{k=1}^{t}\sum_{w\in V}(P^{t-k+1})_{vw}(I-\eta\widehat{C}_M)^{t-k}\{\widehat{C}_M-\widehat{C}_M^{(w)}\}\gamma_{k,w},$$

means the difference is written as

$$\gamma_{t+1,v}-\gamma'_{t+1,v}=\eta\sum_{k=1}^{t}\sum_{w\in V}(P^{t-k+1})_{vw}(I-\eta\widehat{C}_M)^{t-k}\{\widehat{C}_M-\widehat{C}_M^{(w)}\}\gamma_{k,w}.$$

To analyse the difference $\gamma_{t+1,v}-\gamma'_{t+1,v}$ *we then consider the following decomposition where we denote the network averaged iterates* $\overline{\gamma}_t=\frac{1}{n}\sum_{w\in V}\gamma_{t,w}$

$$\|C_M^{1/2}(\gamma_{t+1,v}-\gamma'_{t+1,v})\| \leq \underbrace{\|C_M^{1/2}(\gamma_{t+1,v}-\overline{\gamma}_{t+1})\|}_{\textit{Term 1}}+\underbrace{\|C_M^{1/2}(\overline{\gamma}_{t+1}-\gamma'_{t+1})\|}_{\textit{Term 2}} \tag{14}$$

It is clear the network average can be written using the fact that the communication matrix P *is doubly stochastic i.e.* $\sum_{v\in V}P_{vw}^{t-k+1}=1$ *as follows*

$$\overline{\gamma}_{t+1}-\gamma'_{t+1}=\frac{1}{n}\sum_{v\in V}\gamma_{t+1,v}-\gamma'_{t+1}=\eta\sum_{k=1}^{t}\frac{1}{n}\sum_{w\in V}(I-\eta\widehat{C}_M)^{t-k}\{\widehat{C}_M-\widehat{C}_M^{(w)}\}\gamma_{k,w}.$$

When taking the difference we then arrive at

$$\gamma_{t+1,v} - \gamma'_{t+1} - (\overline{\gamma}_{t+1} - \gamma'_{t+1}) = \eta \sum_{k=1}^{t} \sum_{w \in V} ((P^{t-k+1})_{vw} - \frac{1}{n})(I - \eta \widehat{C}_M)^{t-k} \{\widehat{C}_M - \widehat{C}_M^{(w)}\} \gamma_{k,w}$$

*We can then bound **Term 1** with $\lambda_1 > 0$*

$$\begin{aligned}
&\|C_M^{1/2}(\gamma_{t+1,v} - \overline{\gamma}_{t+1})\| \\
&\leq \eta \sum_{k=1}^{t} \sum_{w \in V} |(P^{t-k+1})_{vw} - \frac{1}{n}| \|C_M^{1/2}(I - \eta \widehat{C}_M)^{t-k} C_{M,\lambda_1}^{1/2}\| \|C_{M,\lambda_1}^{-1/2} \{\widehat{C}_M - \widehat{C}_M^{(w)}\}\| \|\gamma_{k,w}\| \\
&\leq 2\eta \Delta_{\lambda_1} \|N\| \sum_{k=1}^{t} \|C_M^{1/2}(I - \eta \widehat{C}_M^{1/2})^{t-k} C_{M,\lambda_1}^{1/2}\| (\sum_{w \in V} |(P^{t-k+1})_{vw} - \frac{1}{n}|) \\
&\leq 4\eta \Delta_{\lambda_1} \|N\| \sum_{k=1}^{t} \|C_M^{1/2}(I - \eta \widehat{C}_M)^{t-k} C_{M,\lambda_1}^{1/2}\| (\sigma_2^{t-k+1} \wedge 1)
\end{aligned}$$

where we have used that $\|\gamma_{s+1,v}\| \leq \sum_{w \in V} P_{vw} \|(I - \eta \widehat{C}_M^{(w)}) \gamma_{s,w}\| \leq \sum_{w \in V} P_{vw} \|\gamma_{s,w}\| \leq \|N\|$ as well as

$$\begin{aligned}
\|C_{M,\lambda_1}^{-1/2}(\widehat{C}_M - \widehat{C}_M^{(w)})\| &\leq \|C_{M,\lambda_1}^{-1/2}(\widehat{C}_M - C_M)\| + \|C_{M,\lambda_1}^{-1/2}(C_M - \widehat{C}_M^{(w)})\| \\
&\leq \frac{1}{n} \sum_{v \in V} \|C_{M,\lambda_1}^{-1/2}(C_M - \widehat{C}_M^{(v)})\| + \|C_{M,\lambda_1}^{-1/2}(C_M - \widehat{C}_M^{(w)})\| \\
&\leq 2\Delta_{\lambda_1}
\end{aligned}$$

in addition to Lemma 5 to bound $\sum_{w \in V} |(P^{t-k+1})_{vw} - \frac{1}{n}| = \sum_{w \in V} |\Delta^{t-k+1}(v, w)|$.

*To bound **Term 2** we note that we can rewrite*

$$\overline{\gamma}_{t+1} - \gamma'_{t+1} = \eta \sum_{k=2}^{t} \frac{1}{n} \sum_{w \in V} (I - \eta \widehat{C}_M)^{t-k} \{\widehat{C}_M - \widehat{C}_M^{(w)}\} (\gamma_{k,w} - \overline{\gamma}_k).$$

where $\frac{1}{n} \sum_{w \in V} (I - \eta \widehat{C}_M)^{t-k} \{\widehat{C}_M - \widehat{C}_M^{(w)}\} \overline{\gamma}_k = 0$ for $k \geq 1$. Applying triangle inequality as well as similar step to previously, we get with $\lambda_2, \lambda_3 \geq 0$

$$\begin{aligned}
\|C_M^{1/2}(\overline{\gamma}_{t+1} - \gamma'_{t+1})\| &\leq \eta \sum_{k=2}^{t} \|C_M^{1/2}(I - \eta \widehat{C}_M)^{t-k} C_{M,\lambda_2}^{1/2}\| \frac{1}{n} \sum_{w \in V} \|C_{M,\lambda_2}^{-1/2}(\widehat{C}_M - \widehat{C}_M^{(w)})\| \|\gamma_{k,w} - \overline{\gamma}_k\| \\
&\leq 8\eta^2 \Delta_{\lambda_2} \Delta_{\lambda_3} \|N\| \sum_{k=2}^{t} \sum_{\ell=1}^{k-1} \|C_M^{1/2}(I - \eta \widehat{C}_M)^{t-k} C_{M,\lambda_2}^{1/2}\| \|(I - \eta \widehat{C}_M)^{k-1-\ell} C_{M,\lambda_3}^{1/2}\| (\sigma_2^{k-\ell} \wedge 1)
\end{aligned}$$

*where we plugged in the bound from **Term 1** for the deviation $\|\gamma_{k,w} - \overline{\gamma}_k\|$ for $k \geq 2$.*

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

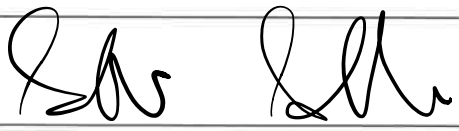
Title of Paper	Decentralised Learning with Random Features and Distributed Gradient Descent
Publication Status	☐Published
Publication Details	Decentralised Learning with Random Features and DistributedGradient Descent", Dominic Richards, Patrick Rebeschini and Lorenzo Rosasco. In International Conference on Machine Learning, 2020

Student Confirmation

Student Name:	Dominic Richards		
Contribution to the Paper	Formulated the main idea and derived technical results. Wrote first draft of manuscript, with later versions written alongside Patrick Rebeschini and Lorenzo Rosasco. Wrote first draft of response to reviewers and implemented feedback. Coded experiments.		
Signature		Date	04/01/2021

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	PATRICK REBESCHINI ASSOCIATE PROFESSOR		
Supervisor comments			
Signature		Date	11/01/2021

This completed form should be included in the thesis, at the end of the relevant chapter.

5

Tree-Based Multi-Task Sparse Recovery with Total Variation Penalty

Tree-Based Multi-Task Sparse Recovery with Total Variation Penalty

Dominic Richards[1] Sahand Negahban[2] Patrick Rebeschini[1]

December 29, 2020

Abstract

Motivated by the setting of distributed multi-task learning, we consider the basic problem of simultaneously recovering a collection of sparse signals from distributed measurements whose similarity structure is encoded in tree-graph topology. We analyse the case where each node is associated with finding the sparse solution of an under-determined system of linear equations, and edges join two nodes if the difference of their solutions is also sparse. We propose a method based on Basis Pursuit Denoising with a total variation penalty, and provide finite sample guarantees for sub-Gaussian matrices. Taking the root of the tree as a reference node, we show that if the sparsity of the differences across nodes is smaller than the sparsity at the root, then recovery is successful with fewer samples than by solving the problems independently, or by using methods that rely on a large overlap in the signal supports, such as the group Lasso. We consider both the noiseless and noisy setting, and numerically investigate the performance of distributed methods based on Distributed Alternating Direction Methods of Multipliers (ADMM) and hyperspectral unmixing.

1 Introduction

Signal processing and machine learning increasingly work with high-dimensional datasets where the number of covariates exceeds the number of samples. To be both statistically and computationally efficient in this setting, it is important to develop approaches that can exploit the structure within the data. A natural assumption in this case is that the data is sparse in some sense. For instance, in compressed sensing [16], the data is assumed to be generated from a sparse signal. Meanwhile in statistics, a subset of features is assumed to be responsible for determining the outcome of interest.

Graphs provide a flexible way to represent relationships between data. In distributed machine learning, graphs are particularly convenient as they can encode both the network used by devices to communicate, as well as potential statistical correlations between data assigned to each device. Tree graphs, in particular, are then fundamental primitives which are often used to understand the behaviour of more general graphs. For instance, spanning trees have recently been used used to prove lower bounds for distributed optimisation algorithms [48, 49] and approximating graph metrics [2, 19], the latter of which has lead to efficient linear time algorithms for both total variation denoising [29, 33, 44] and solving linear equations with Laplacian matrices [51, 57].

[1]Department of Statistics, University of Oxford, 24-29 St Giles', Oxford, OX1 3LB
[2]Department of Statistics and Data Science, Yale University, 24 Hillhouse Ave., New Haven, CT 06510

In this work we consider a collection of standard linear sparse recovery problems, or tasks, that are related through a tree graph. Specifically, each node is associated with the task of finding the sparse solution to a system of under determined linear equations, and an edge joins two nodes (or tasks) if the difference between the solutions to their equations is also sparse. We aim to find a sparse solution by jointly solving these systems of linear equations whilst minimising an appropriate norm across the problems.

Previous works [55, 54, 31, 64, 41, 40, 42, 28] that have considered joint penalisation schemes over a collection of sparse models typically assume that the support associated to each node satisfies an incoherence condition independent of the other nodes [40, 42, 28]. This assumption leads to sub-optimal sample complexities in our setting as the size of the support associated to each node can be large from accumulating the differences associated to the edges in the graph. This arises from the group lasso penalty [55, 64, 28] postulating a *simultaneous sparse* structure where models share non-zero co-ordinates, and thus, the matrix of coefficients is block-sparse. Such an assumption is not implied when the differences between model coefficients are sparse. Our work therefore addresses the following question, which does not seem to have been previously considered in the literature: Can we prove statistical savings if nodes do not satisfy an incoherence conditions with their support?

1.1 Our Contribution

We consider a total variation scheme that penalises differences between nodes that share an edge. This scheme encodes the intuition that if the signal differences are sufficiently sparse then, to recover all signals in the graph, it is more statistically efficient to first recover a single signal associated to a particular reference node (root) and then recover the signal differences associated to edges. Following the Basis Pursuit algorithm [6], we consider the solution that minimises the ℓ_1 norm of the model associated to a root node of the tree and the differences between models that share an edge. A noisy variant similar to Basis Pursuit Denoising [11] is also considered, where the linear constraint is substituted for a bound on the ℓ_2 norm of the residuals. The latter approach being applied within the context of hyperspectral data [27, 17].

Given this framework and assuming sub-Gaussian matrices, we show that statistical savings can be achieved by jointly solving a collection of sparse recovery problems as opposed to solving them either independently or with methods that consider the union of supports. In the noiseless setting, we show that statistical savings are achieved provided the sparsity of the differences is smaller than the sparsity of the solution associated to the root node divided by the square of the number of nodes (Theorem 1). In the noisy setting, we show that the ℓ_1 norm of the estimation error grows at most with the square root of the number of nodes, as opposed to growing linearly if the signals associated to the root node and differences along the edges are solved independently with Basis Pursuit Denoising (Theorem 2).

To the best of our knowledge, our work is the first to demonstrate that statistical savings can be achieved from nodes not needing to satisfy an incoherence conditions with their support. Our results provide theoretical support for a number of applications of the total variation penalty in the context of joint sparse recovery [27, 17]. From an optimisation perspective, we show that in the noiseless case the problem is amenable to a distributed machine learning implementation. Specifically, we show that the objective can be reformulated into a consensus optimisation problem with constraints that reflect the graph topology, and thus, a Distributed Alternating Direction Methods of Multipliers (ADMM) algorithm [4] can be applied. We support our theoretical findings with numerical experiments (Section 3.3) which show the total variation approach can outperform

group lasso methods [28] when the differences between models is assumed sparse, as well as provide qualitative improvements in hyperspectral unmixing with the real-world AVIRIS Cuprite mine data set.

To prove our theoretical results we show that the jointly penalised problem can be reformulated in terms of a standard basis pursuit problem with an augmented matrix and support set. This allows us to leverage the classical Restricted Null Space Property to show that the solution is unique and sparse. To show that the Restricted Null Space Property holds, we exploit the structure of the augmented matrix in conjunction with the Restricted Isometry Property (RIP) of the matrices associated to each task. We show it suffices that the Restricted Isometry Constants of the tasks satisfy two properties. Firstly, the Restricted Isometry Constant of any task should be small with respect to the number of tasks and should hold up to the sparsity of the differences. Secondly, the Restricted Isometry Constant of the root node should hold up to the sparsity of its own signal. Provided the sparsity of the differences is sufficiently small, these two conditions are weaker than requiring Restricted Isometry Constants or incoherence conditions for every node to hold up to the sparsity of their own signals. This yields statistical savings in the case of sub-Gaussian matrices, as the number of samples scales with the sparsity required by each of these. In the noisy setting, we use that a Robust Null Space Property implies bounds on the ℓ_1 estimation error [10, 21]. This can then be shown to hold by using techniques similar to the ones used for the noiseless setting.

1.2 Related Literature

Simultaneously recovering a collection of sparse signals from multiple measurement vectors [18] has been theoretically investigated when performing a form of ℓ_1/ℓ_q regularisation for $q > 1$. Specifically, ℓ_1/ℓ_∞ was investigated within [63, 40, 55] and ℓ_1/ℓ_2 in [36, 42]. Other variants include the dirty model of [28], multi-level lasso of [37] and tree-guided graph lasso of [30]. In the same context, a number of works have investigated variants of greedy pursuit style algorithms [20, 13, 15, 54]. All these methods assume a large overlap between the signals, with the analysis for the group lasso typically assuming each task satisfies an incoherence condition with their own support [40, 28, 42]. In the setting of this work each task's support can become large from accumulating the discrete signals differences associated to edges in the graph (for precise comparison see paragraph in Section 2.3).

The total variation penalty is linked with the fused lasso [61, 25, 53, 9, 47] and has been widely applied to images due to it promoting piece-wise continuous signals which avoids blurring. As far as we are aware, the only work theoretically investigating the total variation penalty as a tool to link a collection of sparse linear recovery problems has been [12]. This work considers the penalised noisy setting and gives both asymptotic statistical guarantees and an optimisation algorithm targeting a smoothed objective. In contrast, we give finite sample guarantees as well as settings where statistical savings are achieved. The application of hyperspectral unmixing [26, 27, 17] has successfully applied the total variation penalty in a manner matching this work. Here, each pixel in an image can be associated to its own sparse recovery problem, for instance, the presence of minerals [27] or the ground class e.g. trees, meadows etc. [17]. It is then natural for the signals to be spatially correlated, and thus, consider the total variation penalty to minimise blurring across the image.

A growing body of works have investigated multi-task learning [8] in distributed contexts. We highlight those most relevant to our setting. The works [60, 56] have considered models penalised in an ℓ_2 sense according to the network topology to encode prior information. The ℓ_2 penalty is not appropriate for the sparse setting of our work. A number of distributed algorithms have been

developed for the sparse setting, for a full review we refer to [1]. The works [28, 52, 59, 35, 43] have developed distributed algorithms that following the group lasso setting, in that, the signals are assumed to be composed of a common shared component plus an individual component. Within [28, 52, 59] this then requires each node to satisfy an incoherence condition, while the setting in [35, 43] is a specific case of a star topology within our work. The work [45] develops a manifold lifting algorithm to jointly recover signals in the absence of an incoherence assumption, although no theoretical guarantees are given.

We highlight the field of federated machine learning [34, 50], where a central node (root) holds a global model, while other devices collect data and update their model with the root whilst accounting for potentially heterogeneity in population distributions across devices. This application fits our setting, as clusters of hierarchically linked computing nodes form tree topologies. For instance, a large server node at the root connected to a collection of medium size server nodes, each of which is itself connected to a number of client nodes.

2 Noiseless Setting

This section formalises the setting that we consider and present our main theoretical results. Section 2.1 introduces the standard problem of sparse recovery with Basis Pursuit. Section 2.2 introduces the Tree-Based Sparse Recovery setting as well as the Total Variation Basis Pursuit problem. Section 2.3 presents our main theoretical result. Section 2.4 presents the main steps in the proof of this result.

2.1 Sparse Recovery with Basis Pursuit

Suppose $x^\star \in \mathbb{R}^p$ is a sparse signal supported on a set $S \subseteq \{1, \dots, p\}$ that is smaller than the dimension, $|S| = s < p$. Define the support of $x^\star$ by the indexes of its non-zero entries. We wish to recover the signal through a matrix $A \in \mathbb{R}^{N \times p}$ and a vector of responses $y \in \mathbb{R}^N$ that satisfy $Ax^\star = y$. The integer N refers to the sample size. To recover the signal we consider the Basis Pursuit Program

$$\min \|x\|_1 \text{ subject to } Ax = y, \tag{1}$$

which is a convex relaxation of the equivalent ℓ_0 penalised problem. It has been shown that, for any $x^\star$ supported on $S \subseteq \{1, \dots, p\}$ the solution to (1) is unique and satisfies $x = x^\star$ if and only if A satisfies the Restricted Null Space property with respect to S, see for instance [14], that is,

$$2\|x_S\|_1 \leq \|x\|_1 \quad \text{ for any } x \in \mathrm{Ker}(A) \backslash \{0\}. \tag{2}$$

The Restricted Isometry Property (RIP) [7] is a sufficient condition for A to satisfy the Restricted Null Space Property. Precisely, a matrix A satisfies RIP at sparsity level s if there exists a constant $c_s \geq 0$ such that $(1-c_s)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1+c_s)\|x\|_2^2$ for all $\|x\|_0 \leq s$. If A has independent and identically distributed (i.i.d.) sub-Gaussian entries, that is, if the i,jth entry A_{ij} satisfies $\mathbb{P}(|A_{ij}| \geq t) \leq \beta e^{-\kappa t^2}$ for all $t \geq 0$ for sub-Gaussian parameters β and κ, then with probability at least $1-\epsilon$, the matrix $A/\sqrt{N}$ has Restricted Isometry Constant upper bounded as $c_s \leq \delta$ if the sample size N satisfies $N \geq C\delta^{-2}(s \log(ep/s) + \log(2/\epsilon))$ for some constant $C > 0$ (Theorem 3 in Appendix D).

2.2 Tree-Based Sparse Recovery

Suppose we have a collection of $|V| = n$ nodes (tasks or agents) connected by a tree graph $G = (V, E)$ with edges $E \subseteq V \times V$. Each node $v \in V$ wishes to recover a sparse vector $x_v^\star \in \mathbb{R}^p$ supported on $S_v \subseteq \{1, \dots, p\}$. We index the nodes from the set $\{1, \dots, n\}$ where 1 represents the root of the tree. We assume the signal sparsity of the root node is $s \in [1, p]$ i.e. $|S_1| \leq s$. Following the standard sparse recovery problem described in Section 2.1, each agent has a matrix $A_v \in \mathbb{R}^{N_v \times p}$ as well as response $y_v \in \mathbb{R}^{N_v}$ for which the sparse signal associated to them satisfies $A_v x_v^\star = y_v$. The natural number $N_v > 0$ is then the number of samples held by agent $v \in V$.

We consider the setting where the signal held by nodes in the network are related. Specifically, we assume that for every edge $e = \{v, w\} \in E$ in the network, the difference between the signals associated to the nodes $x_v^\star - x_w^\star$ is also sparse, that is, supported on the set $S_e = S_{\{v,w\}} \subseteq \{1, \dots, p\}$, which we assume is at most of the size $|S_e| < s'$. We assume that the sparsity of the difference of neighbours signals s' is smaller than the sparsity of the signals held by the root node, namely $s' < s$.

Tree Baseline and Stepwise Approach. Suppose for any pair of edges $e, e' \in E$ the supports of the differences are disjoint from each other $S_e \cap S_{e'} = \emptyset$ as well as from the support of the root $S_e \cap S_1 = \emptyset$. Moreover, suppose G is a tree of depth $0 \leq D \leq n$ and let the integer $i_v \in \{0, \dots, D\}$ denote the depth of node $v \in V$ in the tree i.e. the distance from the root. Following the discussion in Section 2.1, if each node has sub-Gaussian matrices and performed Basis Pursuit independently, then the number of samples required by agent v would then in this case scale as $N_v \geq s + i_v s'$. In the case of a path topology with the root at one end, where the depth is $D = n - 1$ and i_v is the index of node v along the path, the total sample complexity is then least $\sum_{v \in V} N_v \geq ns + n(n-1)s'/2$. Although a more natural total sample complexity of $s + ns'$ can be achieved by proceeding in a stepwise manner as follows. Order s samples can recover the root signal, while order $n \times s'$ samples can recover each of the differences associated to the edges. Any nodes signal can then be recovered by summing up the differences along the edges. This yields a saving from approximately $ns + n^2 s'$ to $s + ns'$, which would be significant when the sparsity of the difference s' is small and the network size n is large. This embodies the main intuition for the statistical savings that we set to unveil in our work.

Distributed Machine Learning. Tree topologies naturally arise in distributed machine learning when considering layers of computing clusters. In the simplest case of a star topology, where a central server (root) is connected to individual clients, the tree has depth $D = 1$. Meanwhile, more generally, D layers of computing clusters results in a tree of depth D. As we have seen, the size of the support for the tasks furthest from the root is then on the order of $s + (D-1)s'$. Therefore, when considering variants of the group lasso like the dirty model of [28], some tasks require sample sizes to scale as $N_v \geq s + (D-1)s'$ to ensure their matrices satisfy an incoherence condition for their support sets. That is, a matrix $B \in \mathbb{R}^{N \times p}$ satisfies incoherence condition for support $U \subseteq \{1, \dots, p\}$ if $\max_{j \in U^c} \|B_j^\top B_U (B_U^\top B_U)^{-1}\|_1$ is finite, where B_j is the jth column of B and B_U is the matrix B restricted to columns with indices in U. To ensure $B_U^\top B_U$ can be inverted we then require $N \geq |U|$.

Multiple Measurement Vector Framework. The greedy pursuit style algorithms [20, 13, 15, 54] in the Multiple Measurement Vector Framework [18] make the assumption that each task's Restricted Isometry Constants hold up to the size of the union of supports. In the Tree-Based Sparse Recovery setting the union of supports can grow up to $s + ns'$ for any topology, as such, each task would require $N_v \geq s + ns'$ samples in this case.

2.3 Main Theoretical Result. Guarantees for Total Variation Basis Pursuit

We provide a setting which achieves the goals outlined in Section 2.2. We encode the assumptions described in the Tree-Based Sparse Recovery setting described in Section 2.2 in the following optimisation problem, which we call *Total Variation Basis Pursuit*:

$$\min_{x_1,\dots,x_n} \|x_1\|_1 + \sum_{e=\{v,w\}\in E} \|x_v - x_w\|_1 \text{ subject to}$$
$$A_v x_v = y_v \quad \forall v \in V. \tag{3}$$

This problem aims to simultaneously recover an individual signal, specifically the signal associated to the root node, through the penalisation $\|x_1\|_1$, as well as the differences associated to edges in the graph, through the total variation penalty $\sum_{e=\{v,w\}\in V} \|x_v - x_w\|_1$. This is subject to each node satisfying their linear constraint $A_v x_v = y_v$. The following theorem then gives sufficient conditions on the number of samples for the solution of (3) to recover the signals associated to each node.

Theorem 1. *Consider the Tree Based Sparse Recovery setting in Section 2.2. Suppose the matrices $\{A_v\}_{v\in v}$ are independent and have i.i.d sub-Gaussian entries. Fix any $\epsilon > 0$. Then, with probability greater than $1-\epsilon$, for any collection of signals $\{x_v^\star\}_{v\in V}$, the solution to the Total Variation Basis Pursuit problem (3) is unique and satisfies $x_v = x_v^\star$ for all $v \in V$ provided*

$$\underbrace{N_1 \gtrsim s\big(\log(p/s) + \log(1/\epsilon)\big)}_{\textit{Root node samples}} \text{ and } \underbrace{N_v \gtrsim n^2 s'\big(\log(p/s') + \log(n/\epsilon)\big) \quad \textit{for all } v \in V}_{\textit{All node samples}}.$$

We now discuss Theorem 1. The root node's sample size N_1 is required to grow with the sparsity of its own signal s, while every nodes samples size is required to scale as $N_v \geq n^2 s'$ for $v \in V$. This yields a total sample complexity of the order $s + n^3 s'$, while the stepwise approach in Section 2.2 requires order $s + ns'$ samples, and thus, there is a factor n^2 worse in front of s'. One possible reason for this difference is that (3) recovers the signals simultaneously, while the process in Section 2.1 recovers the signals (and differences) in a stepwise manner. Furthermore, each signal is sensed through a potentially different matrix A_v so the response (if $N_v = N_w$) differences $y_v - y_w$ do not align with the signal differences $x_v - x_w$. We leave investigating these aspects to future work. We note if the root node were to change to a node distance k from the original root, then for result of Theorem 1 to hold the new root would require order $s + ks'$ samples. Regarding an application to distributed machine learning, a distributed ADMM algithm for solving the Total Variation Basis Pursuit problem is investigated in Appendix A.2. We note that, empirically, the algorithm is observed to converge at a linear rate.

Distributed Machine Learning. Here we describe when the sample conditions in Theorem 1 yield savings for distributed machine learning, described in Section 2.2. Recall, the methods discussed require agent's sample size to scale with the sparsity of the root s. Meanwhile, Total Variation Basis Pursuit requires non-root agents to hold $n^2 s'$ samples. Therefore, savings are achieved for non-root agents when the sparsity of the differences satisfies $s' \leq s/n^2$. This is relevant for distributed machine learning as client devices may have small amounts of memory. Savings in total sample complexity are also achieved when $s' \leq s/n^2$, as the previous methods require at least a total of $ns + ns'$ samples, while Total Variation Basis Pursuit requires $s + n^3 s'$. The sample savings for the non-root agent are supported by the experiments presented in Figure 2 in Appendix A.1. The experiments suggest that sample savings hold for larger numbers of agents n than what our analysis suggests, i.e. $s' \leq s'/n^2$.

From Trees to General Topologies. While our theoretical results consider tree topologies, in a similar manner to total variation denoising [44], more general topologies can be considered. In particular, if the signal is sparse with respect to a graph, then it is sparse with respect to *any* spanning tree of that graph. Therefore, for a general graph we can follow a two step procedure. Firstly, construct a spanning tree of the general graph. Secondly, solve Total Variation Basis Pursuit (3) with this spanning tree.

2.4 Proof of Theorem 1

This section gives the two main steps of the proof for Theorem 1. The first step is to reformulate the Total Variation Basis Pursuit problem (3) in terms of a standard Basis Pursuit problem (2.1) with a particular matrix A and sparsity set. The second step shows the Restricted Null Space Property can hold for the reformulated problem. The steps are outlined in the following two sections.

2.4.1 Reformulating Total Variation Basis Pursuit as Standard Basis Pursuit

We begin by introducing some notation. For node $v \in V$, denote the set of edges making a path from node v to the root node 1 by $\pi(v) = \{\{v, w_1\}, \{w_1, w_2\}, \ldots, \{w_{k_v-1}, w_{k_v}\}, \{w_{k_v}, 1\}\} \subseteq E$ where $k_v \geq 1$ is the number of intermediate edges. In the case $k_v = 0$ there is only a single edge and so we write $\pi(v) = \{v, 1\} \in E$. Meanwhile, for the root node itself $v = 1$ we simply have the singleton $\pi(v) = \pi(1) = \{1\}$, and thus, we have the root node included $v \in \pi(v)$ but no edges i.e. $e \notin \pi(v)$ for any $e \in E$. For each edge $e = \{v, w\} \in E$ the difference is denoted $\Delta_e = x_v - x_w$, and so the vector associated to any node x_v can be decomposed into the root node x_1 plus the differences along the path $x_v = x_1 + \sum_{e \in \pi(v)} \Delta_e$. Similarly, the signal associated to each node $x_v^\star$ can be decomposed into differences of signals associated to the edges $e = \{v, w\} \in E$ with $\Delta_e^\star = x_v^\star - x_w^\star$.

With this notation we can then reformulate (3) in terms of x_1 and $\{\Delta_e\}_{e \in E}$ as follows

$$\min_{x_1, \{\Delta_e\}_{e \in E}} \|x_1\|_1 + \sum_{e=(v,w) \in E} \|\Delta_e\|_1 \text{ subject to} \qquad (4)$$
$$A_v\Big(x_1 + \sum_{e \in \pi(v)} \Delta_e\Big) = y_v \quad \forall v \in V.$$

Optimisation problem (4) is now in terms of a standard basis pursuit problem (1) with, if edges are labeled with integers, the vector $x = (x_1, \Delta_1, \ldots, \Delta_{|E|})$, true signal $x^\star = (x_1^\star, \Delta_1^\star, \ldots, \Delta_{|E|}^\star)$, and a matrix A. To be precise, the matrix A can be defined in terms of blocks $A = (H_1^\top, \ldots, H_n^\top)^\top \in \mathbb{R}^{(\sum_{v \in V} N_v) \times np}$ with each block $H_v \in \mathbb{R}^{N_v \times np}$ for $v \in V$. Each block then defined as $H_v = \Big(H_{v1}, H_{v2}, \ldots, H_{vn}\Big)$ with, for $i = 1, \ldots, n$, the matrix $H_{vi} = A_v$ if node i is included on the path going from node v to the root node 1 i.e. $i \in \pi(v)$, and 0 otherwise.

The signal associated to the reformulated problem (4) remains sparse and is supported on a set S with a particular structure due to encoding the sparsity of the differences $\{\Delta_e^\star\}_{e \in E}$. Specifically, the set S contains the entries from $\{1, \ldots, p\}$ aligned with S_1 and, labeling the edges $e \in E$ with the integers $i = 1, \ldots, |E|$, the elements from $\{1, \ldots, p\}$ associated to S_e offset by $i \times p$. Now that (3) is in terms of a Basis Pursuit problem, its success relies on the matrix A satisfying the Restricted Null Space Property (2) with respect to the sparsity set S.

2.4.2 Restricted Null Space Property for Reformulated Problem

To show the Restricted Null Space Property (2) is satisfied with the matrix A and sparsity set S as defined in Section 2.4.1, we consider the Restricted Isometry Constants of the matrices $\{A_v\}_{v\in V}$. Let $\delta_k^{(1)} \in (0,1)$ denote the Restricted Isometry Constant for the root node matrix A_1 at the sparsity level k. Meanwhile, let $\delta_k \in (0,1)$ denote the largest Restricted Isometry Constant of all the matrices $\{A_v\}_{v\in V}$.

The proof begins by utilising the linear constraints in (4) to control, for any node $v \in V$, the norm of the vector $x_1 + \sum_{e\in\pi(v)} \Delta_e$ restricted to small sets of size at most s'. Meanwhile, control of x_1 for larger sets up-to the size s is done through A_1. This is summarised in the following lemma.

Lemma 1. *With $x = (x_1, \Delta_1, \dots, \Delta_{|E|}) \in Ker(A)\backslash\{0\}$ and A as in (4) we have for all $v \in V$ and $U \subseteq \{1,\dots,p\}$ such that $|U| \le s'$*

$$\big\|(x_1 + \sum_{e\in\pi(v)} \Delta_e)_U\big\|_1 \le \frac{\delta_{2s'}}{1-\delta_{2s'}} \Big\|x_1 + \sum_{e\in\pi(v)} \Delta_e\Big\|_1. \tag{5}$$

Furthermore, for all $U \subseteq \{1,\dots,p\}$ such that $|U| \le s$

$$\|(x_1)_U\|_1 \le \frac{\delta_{2s}^{(1)}}{1-\delta_{2s}^{(1)}} \|x_1\|_1. \tag{6}$$

The proof of Lemma 1 can be found in Appendix D. It follows from techniques typically used for showing that the Restricted Null Space Property holds in the context of Basis Pursuit applied for each matrix $\{A_v\}_{v\in V}$ aligning with the optimality conditions in (4).

For $x \in \mathrm{Ker}(A)\backslash\{0\}$ we set to upper bound $\|(x)_S\|_1 = \|(x_1)_{S_1}\|_1 + \sum_{e\in E} \|(\Delta_e)_{S_e}\|_1$ by the ℓ_1 norm $\|x\|_1$. We note it suffices to let S_1 be the indices of the largest s elements of x_1, and, for each $e \in E$, the set S_e be the indices of the largest s' elements of Δ_e. From Lemma 1, equation (6), we immediately get the upper bound $\|(x_1)_{S_1}\| \le \delta_{2s}^{(1)}\|x\|_1/(1-\delta_{2s}^{(1)})$. For $e = \{v,w\} \in E$ consider $\|(\Delta_e)_{S_e}\|_1$. Suppose w is the node on the edge e closest to the root node. If not, simply swap the labels. By adding and subtracting $\big(x_1 + \sum_{\widetilde{e}\in\pi(w)} \Delta_{\widetilde{e}}\big)_{S_e}$ we get

$$\|(\Delta_e)_{S_e}\|_1 \le \Big\|(x_1 + \sum_{\widetilde{e}\in\pi(w)} \Delta_{\widetilde{e}})_{S_e}\Big\|_1 + \Big\|(x_1 + \sum_{\widetilde{e}\in\pi(v)} \Delta_{\widetilde{e}})_{S_e}\Big\|_1 \le \frac{2\delta_{2s'}}{1-\delta_{2s'}}\|x\|_1,$$

where on the first equality we used that $\big(x_1 + \sum_{\widetilde{e}\in\pi(w)} \Delta_{\widetilde{e}}\big)_{S_e} + (\Delta_{\widetilde{e}})_{S_e} = \big(x_1 + \sum_{\widetilde{e}\in\pi(v)} \Delta_{\widetilde{e}}\big)_{S_e}$ since the edge $e = \{v,w\}$ is included onto the path from node w, i.e. $\pi(w)$, thus making it a path from node v i.e. $\pi(v)$. The second inequality comes from Lemma 1 (5). Summing up the above for all $e \in E$ and adding the previous bound for $\|(x_1)_{S_1}\|_1$, we get $\|(x)_S\|_1 \le 4\big(\frac{\delta_{2s}^{(1)}}{1-\delta_{2s}^{(1)}} \vee \frac{n\delta_{2s'}}{1-\delta_{2s'}}\big)\|x\|_1$. The Restricted Null Space Property (2) then holds when $\delta_{2s}^{(1)} < 1/9$ and $\delta_{2s'} < 1/9n$.

To ensure that the conditions on the Restricted Isometry Constants $\delta_{2s}^{(1)}$ and $\delta_{2s'}$ are satisfied when the entries of $\{A_v\}_{v\in V}$ are i.i.d. sub-Gaussian, we recall the statement at the end of Section 2.1. For the Restricted Isometry Constant for node 1 to be upper bounded $\delta_{2s}^{(1)} \le 1/9$ with probability greater than $1-\epsilon$, it is sufficient to have $N_1 \ge 81C(2s\log(ep/2s) + \log(2/\epsilon)$ (see Theorem 3 in Appendix D). Meanwhile, for the maximum Restricted Isometry Constant across all agents to be upper bounded $\delta_{2s'} \le 1/(9n)$ with probability greater than $1-\epsilon$, it is sufficient to take a union bound across the agents with $\min_{v\in v} N_v \ge C(81n^2)(2s\log(ep/s) + \log(2n/\epsilon))$.

3 Noisy Setting

This section studies noisy Tree-Based Sparse Recovery. Section 3.1 introduces Basis Pursuit Denoising. Section 3.2 extends Total Variation Basis Pursuit to consider noise. Section 3.3 presents the experiments.

3.1 Basis Pursuit Denoising

We begin by considering the standard Basis Pursuit problem described in Section 2.1 with two additional extensions. Firstly, the true signal $x^\star$ is not exactly s sparse. Secondly, there is noise so $y \approx Ax^\star$, or, more precisely there exists $\eta > 0$ such that $\|Ax - y\|_2 \leq \eta$. To address this case, we consider the Basis Pursuit Denoising problem [11] which is formulated as

$$\min_x \|x\|_1 \text{ subject to } \|Ax - y\|_2 \leq \eta. \tag{7}$$

Naturally, the equality constraint $Ax = y$ in the noiseless setting has been swapped for an upper bound on the discrepancy $\|Ax - y\|_2$. To investigate guarantees for the solution to (7), we consider the Robust Null Space Property, see for instance [21]. A matrix A is said to satisfy the Robust Null Space Property for a set $S \subseteq \{1, \ldots, p\}$ and parameters $\rho, \tau \geq 0$ if

$$\|x_S\|_1 \leq \rho \|x_{S^c}\|_1 + \tau \|Ax\|_2 \text{ for all } x \in \mathbb{R}^N. \tag{8}$$

Given condition (8), bounds on the ℓ_1 estimation error between a solution to the Denoising Basis Pursuit problem (7) and the true underlying signal $x^\star$ can be obtained. That is, for any solution to (7), $x \in \mathbb{R}^p$ with $y = Ax^\star + e$ where $\|e\|_2 \leq \eta$, we have (see [21, Theorem 4.2] with $z = x^\star$)

$$\|x - x^\star\|_1 \leq \underbrace{\frac{2(1+\rho)}{1-\rho} \|(x^\star)_{S^c}\|_1}_{\text{Sparse Approximation}} + \underbrace{\frac{4\tau}{1-\rho}\eta}_{\text{Noise}}.$$

The first term above encodes that $x^\star$ is not exactly s sparse, while the second term represents error from the noise. We now discuss the values taken by η and τ in the case that A has i.i.d. sub-Gaussian entries. Recall from Section 2.1 that the scaled matrix $A/\sqrt{N}$ in this case can satisfy a Restricted Isometry Property, and thus, it is natural to choose $\eta = \sqrt{N}\eta_{\text{Noise}}$ for $\eta_{\text{Noise}} \geq 0$ since the ℓ_2 bound on the residuals in (7) becomes $\|Ax - y\|_2/\sqrt{N} \leq \eta_{\text{Noise}}$. We can then pick $\|e\|_2/\sqrt{N} \leq \eta_{\text{Noise}}$, which is an upper bound on the standard deviation of the noise. The Robust Null Space Property then holds in this case, see [21, Theorem 4.22], with $\tau \approx \sqrt{s}$, leading to a ℓ_1 error bound of the order $\|x - x^\star\|_1 \lesssim \|(x^\star)_{S^c}\|_1 + \eta_{\text{Noise}}\sqrt{s}$ (see [58, Theorem 7.13]).

3.2 Total Variation Basis Pursuit Denoising

We return to the Tree-Based Sparse Recovery setting as in Section 2.2 to consider the noisy case analogous to the one described in Section 3.1. That is, the root node signal $x_1^\star$ and the $\{\Delta_e^\star\}_{e \in E}$ are approximately sparse and each agent $v \in V$ holds noisy samples $y_v \approx A_v x_v^\star$. Reformulating the Total Variation Basis Pursuit problem into a Basis Pursuit problem (4) and bounding the ℓ_2 norm of the

residuals, then yields the *Total Variation Basis Pursuit Denoising* problem

$$\min_{x_1,\Delta_e\in E} \|x_1\|_1 + \sum_{e\in E}\|\Delta_e\|_1 \text{ subject to} \qquad (9)$$
$$\sum_{v\in V}\|A_v(x_1 + \sum_{e\in\pi(v)}\Delta_e) - y_v\|_2^2 \le \eta^2.$$

Where η^2 now upper bounds the squared ℓ_2 norm of the noise summed across all of the nodes i.e. $\sum_{v\in V}\|A_v x_v^\star - y_v\|_2^2$. This is now in the form of (7) with an augmented matrix A as in Section 2.4.1. The following theorem then gives, in terms of the Restricted Isometry Constants δ_k and $\delta_k^{(1)}$ defined in Section 2.4.2, values for ρ and τ for which this matrix A satisfies the Robust Null Space Property (8).

Theorem 2. *Consider the A matrix and sparsity set S as constructed in Section 2.4. Then A satisfies the Robust Null Space Property with* $\rho = \rho'/(1-\rho')$ *and* $\tau = \tau'/(1-\rho')$ *where*

$$\rho' = 4\Big(\frac{N\delta_{2s'}}{1-\delta_{s'}} \vee \frac{\delta_{2s}^{(1)}}{1-\delta_s^{(1)}}\Big) \quad \textit{and}$$
$$\tau' = \frac{\sqrt{1+\delta_{s'}}}{1-\delta_{s'}} \vee \frac{\sqrt{1+\delta_s^{(1)}}}{1-\delta_s^{(1)}}(\sqrt{s} + Deg(G)\sqrt{ns'}).$$

The parameter ρ' in Theorem 2 appeared in noiseless case to show the Restricted Null Space Property held. Meanwhile, the parameter τ' scales (up to a network degree Deg(G) factor) with the sparsity of the Tree-Based Sparse setting described in Section 2.2. That is, if each agent had i.i.d. sub-Gaussian matrices and we chose $\eta = \sqrt{\sum_{v\in V} N_v}\eta_{\text{Noise}}$ where $\eta_{\text{Noise}} > 0$ upper bounds the noise standard deviation across all of the agents, the ℓ_1 estimation error of the solution to (9) is then of the order

$$\|x_1 - x_1^\star\|_1 + \sum_{e\in E}\|\Delta_e - \Delta_e^\star\|_1 \lesssim \underbrace{\|(x^\star)_{S^c}\|_1}_{\text{Approximate Sparsity}} + \underbrace{(\sqrt{s} + \text{Deg}(G)\sqrt{ns'})\eta_{\text{Noise}}}_{\text{Noise}}.$$

The error scales with the approximate sparsity of the true signal through $\|(x^\star)_{S^c}\|_1$ and now the noise term with the effective sparsity $\sqrt{s} + \text{Deg}(G)\sqrt{ns'}$.

Comparison to Stepwise Approach For the stepwise approach in Section 2.2, where the root node and the edges were estimated independently using Basis Pursuit Denoising, the resulting noise term scales as $\sqrt{s} + n \times \sqrt{s'}$. Therefore solving the Total Variation Basis Pursuit Denoising offers a order $\sqrt{n}$ saving in ℓ_1 estimation error over the step wise approach. This highlights at two sample size regimes. A low sample size setting where the total sample size is order $s + ns'$, the step wise approach is provably feasible and an estimation error on the order of $\sqrt{s} + n\sqrt{s'}$ can be achieved. And a higher sample size setting where the total sample size is $s + n^3 s'$, Total Variation Basis Pursuit Denoising is provably feasible and an estimation on the order of $\sqrt{s} + \sqrt{ns'}$ can be achieved.

3.3 Experiments. Total Variation Basis Pursuit Denoising

This section present simulation results for the Total Variation Basis Pursuit Denoising problem (9). The following paragraphs, respectively, describe results for simulated and real data.

Simulated Data. Figure 4 in Appendix B.1 shows the ℓ_1 estimation error for Total Variation Basis Pursuit Denoising, group lasso and the dirty model of [28]. This is plotted against the number of agents for both path and balanced tree topologies. Observe as the number of agents grows the estimation error for the group lasso methods grows quicker than the total variation approach. The group lasso variants perform poorly here due to the union of supports growing with the number of agents, and thus, the small overlap between agent's supports. The balanced tree topology here is a realistic model for networks of computing clusters.

Hyperspectral Unmixing. We apply Total Variation Basis Pursuit Denoising to the popular AVIRIS Cuprite mine reflectance dataset `https://aviris.jpl.nasa.gov/data/free_data.html` with a subset of the USGS library splib07 [32]. As signals can be associated to pixels in a 2-dimensional image, it is natural here to consider the total variation associated with a grid topology. Although, computing the total variation explicitly in this case can be computationally expensive, see for instance [44]. We therefore simplify the objective in two respects. Firstly, the image is tiled into groups of $n = 4$ pixels arranged in a 2x2 grid, with each group considered independently. This is common approach within parallel rendering techniques, see for instance [38], and is justified in our case as the signals are likely most strongly correlated with their neighbours in the graph. Note that this also allows our approach to scale to larger images as the algorithm can be run on each tile in an embarrassingly parallel manner. Secondly, following the discussion in paragraph **From Trees to General Topologies** in Section 2.3, a spanning tree of the 4 pixels groups is constructed by removing a single edge from the 2x2 grid. More details of the experiment are in Appendix B.2.

We considered four methods: applying Basis Pursuit Denoising to each pixel independently; Total Variation Denoising (9) applied to the groups of 4 pixels as described previously; the group lasso applied to the groups of 4 pixels described previously; and a baseline Hyperspectral algorithm SUNnSAL [3]. Figure 1 then gives plots of the coefficients associated to two minerals for three of the methods. Additional plots associated to four minerals and the four methods have been Figure 7 Appendix B.2. Recall, by combining pixels the aim is to estimate more accurate coefficients than from denoising them independently. Indeed for the Hematite, Andradite and Polyhalite minerals, less noise is present for the total variation approach, alongside larger and brighter clusters. This is also in comparison to SUNnSAL, where the images for Andradite and Polyhalite from the total variation approach have less noise and brighter clusters. Although, we note that combining groups of pixels in this manner can cause the images to appear at a lower resolution.

4 Conclusion

In this work we considered total variation penalty methods to jointly recover a collection of sparse signals related by a tree graph. We assumed a tree-based sparse structure for the signals, where the signal at the root and the signal differences along edges were sparse. This setting differs from previous work on solving collections of sparse problems, which assume large overlapping supports between signals. We demonstrated (in noiseless and noisy settings) that statistical savings can be achieved over these methods as well as solving each problem independently, in addition to developing a distributed ADMM algorithm for solving the objective function in the noiseless case.

Moving forward, a theoretical gap is currently present for the noiseless case, as order $s+ns'$ samples are sufficient if the signals and their differences are recovered in a stepwise manner, while we require $s+n^3s'$ for simultaneous recovery. Following this work, a distributed ADMM algorithm can be developed for noisy signals.

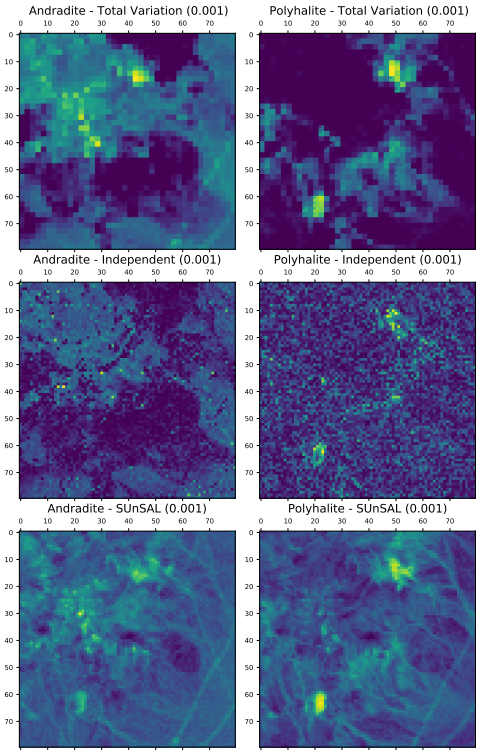


Figure 1: Coefficients associated to the minerals Andradite (*Left*), and Polyhalite (*Right*). Methods are, *Top*: Total Variation Basis Pursuit Denoising applied to 2x2 pixel tiles with $\eta = 0.001$; *Middle*: Basis Pursuit Denoising applied independently to each pixel with $\eta = 0.001$. *Bottom*: SUNnSAL with regularisation of 0.001. Yellow pixels indicate higher values.

References

[1] Ghanbar Azarnia, Mohammad Ali Tinati, and Tohid Yousefi Rezaii. Cooperative and distributed algorithm for compressed sensing recovery in wsns. *IET Signal Processing*, 12(3):346–357, 2017.

[2] Yair Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *Proceedings of 37th Conference on Foundations of Computer Science*, pages 184–193. IEEE, 1996.

[3] José M Bioucas-Dias and Mário AT Figueiredo. Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing. In *2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, pages 1–4. IEEE, 2010.

[4] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011.

[5] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[6] Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(8):1207–1223, 2006.

[7] Emmanuel J Candes and Terence Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.

[8] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[9] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2):89–97, 2004.

[10] Venkat Chandrasekaran, Benjamin Recht, Pablo A Parrilo, and Alan S Willsky. The convex geometry of linear inverse problems. *Foundations of Computational mathematics*, 12(6):805–849, 2012.

[11] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001.

[12] Xi Chen, Seyoung Kim, Qihang Lin, Jaime G Carbonell, and Eric P Xing. Graph-structured multi-task regression and an efficient optimization method for general fused lasso. *arXiv preprint arXiv:1005.3579*, 2010.

[13] Yi Chen, Nasser M Nasrabadi, and Trac D Tran. Hyperspectral image classification using dictionary-based sparse representation. *IEEE transactions on geoscience and remote sensing*, 49(10):3973–3985, 2011.

[14] Albert Cohen, Wolfgang Dahmen, and Ronald DeVore. Compressed sensing and best k-term approximation. *Journal of the American mathematical society*, 22(1):211–231, 2009.

[15] Wei Dai and Olgica Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *IEEE transactions on Information Theory*, 55(5):2230–2249, 2009.

[16] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[17] Peijun Du, Zhaohui Xue, Jun Li, and Antonio Plaza. Learning discriminative sparse representations for hyperspectral image classification. *IEEE Journal of Selected Topics in Signal Processing*, 9(6):1089–1104, 2015.

[18] Marco F Duarte and Yonina C Eldar. Structured compressed sensing: From theory to applications. *IEEE Transactions on signal processing*, 59(9):4053–4085, 2011.

[19] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *Journal of Computer and System Sciences*, 69(3):485–497, 2004.

[20] Joe-Mei Feng and Chia-Han Lee. Generalized subspace pursuit for signal recovery from multiple-measurement vectors. In *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2874–2878. IEEE, 2013.

[21] Simon Foucart and Holger Rauhut. An invitation to compressive sensing. In *A mathematical introduction to compressive sensing*, pages 1–39. Springer, 2013.

[22] Bingsheng He and Xiaoming Yuan. On the o(1/n) convergence rate of the douglas–rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2):700–709, 2012.

[23] BS He, Hai Yang, and SL Wang. Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. *Journal of Optimization Theory and applications*, 106(2):337–356, 2000.

[24] Mingyi Hong and Zhi-Quan Luo. On the linear convergence of the alternating direction method of multipliers. *Mathematical Programming*, 162(1-2):165–199, 2017.

[25] Jan-Christian Hütter and Philippe Rigollet. Optimal rates for total variation denoising. In *Conference on Learning Theory*, pages 1115–1146, 2016.

[26] Marian-Daniel Iordache, José M Bioucas-Dias, and Antonio Plaza. Sparse unmixing of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 49(6):2014–2039, 2011.

[27] Marian-Daniel Iordache, José M Bioucas-Dias, and Antonio Plaza. Total variation spatial regularization for sparse hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 50(11):4484–4502, 2012.

[28] Ali Jalali, Sujay Sanghavi, Chao Ruan, and Pradeep K Ravikumar. A dirty model for multi-task learning. In *Advances in neural information processing systems*, pages 964–972, 2010.

[29] Nicholas A Johnson. A dynamic programming algorithm for the fused lasso and l 0-segmentation. *Journal of Computational and Graphical Statistics*, 22(2):246–260, 2013.

[30] Seyoung Kim and Eric P Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *ICML*, volume 2, page 1, 2010.

[31] Yuwon Kim, Jinseog Kim, and Yongdai Kim. Blockwise sparse regression. *Statistica Sinica*, 16(2):375, 2006.

[32] Raymond F Kokaly, Roger N Clark, Gregg A Swayze, K Eric Livo, Todd M Hoefen, Neil C Pearson, Richard A Wise, William M Benzel, Heather A Lowers, Rhonda L Driscoll, et al. Usgs spectral library version 7. Technical report, US Geological Survey, 2017.

[33] Vladimir Kolmogorov, Thomas Pock, and Michal Rolinek. Total variation on a tree. *SIAM Journal on Imaging Sciences*, 9(2):605–636, 2016.

[34] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

[35] Xiaowei Li. *A weighted -minimization for distributed compressive sensing*. PhD thesis, University of British Columbia, 2015.

[36] Karim Lounici, Massimiliano Pontil, Alexandre B Tsybakov, and Sara Van De Geer. Taking advantage of sparsity in multi-task learning. *arXiv preprint arXiv:0903.1468*, 2009.

[37] Aurelie C Lozano and Grzegorz Swirszcz. Multi-level lasso for sparse multi-task regression. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 595–602. Omnipress, 2012.

[38] Steven Molnar, Michael Cox, David Ellsworth, and Henry Fuchs. A sorting classification of parallel rendering. *IEEE computer graphics and applications*, 14(4):23–32, 1994.

[39] João FC Mota, João MF Xavier, Pedro MQ Aguiar, and Markus Puschel. Distributed basis pursuit. *IEEE Transactions on Signal Processing*, 60(4):1942–1956, 2011.

[40] Sahand Negahban and Martin J Wainwright. Joint support recovery under high-dimensional scaling: Benefits and perils of -regularization. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, pages 1161–1168. Curran Associates Inc., 2008.

[41] Guillaume Obozinski, Ben Taskar, and Michael Jordan. Joint covariate selection for grouped classification. Technical report, Technical Report 743, Dept. of Statistics, University of California Berkeley, 2007.

[42] Guillaume Obozinski, Martin J Wainwright, Michael I Jordan, et al. Support union recovery in high-dimensional multivariate regression. *The Annals of Statistics*, 39(1):1–47, 2011.

[43] Felix Oghenekohwo, Haneet Wason, Ernie Esser, and Felix J Herrmann. Low-cost time-lapse seismic with distributed compressive sensing—part 1: Exploiting common information among the vintages. *Geophysics*, 82(3):P1–P13, 2017.

[44] Oscar Hernan Madrid Padilla, James Sharpnack, James G Scott, and Ryan J Tibshirani. The dfs fused lasso: Linear-time denoising over general graphs. *Journal of Machine Learning Research*, 18:176–1, 2017.

[45] Jae Young Park and Michael B Wakin. A geometric approach to multi-view compressive imaging. *EURASIP Journal on Advances in Signal Processing*, 2012(1):37, 2012.

[46] Marcos Raydan. The barzilai and borwein gradient method for the large scale unconstrained minimization problem. *SIAM Journal on Optimization*, 7(1):26–33, 1997.

[47] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[48] Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. *arXiv preprint arXiv:1702.08704*, 2017.

[49] Kevin Scaman, Francis Bach, Sébastien Bubeck, Laurent Massoulié, and Yin Tat Lee. Optimal algorithms for non-smooth distributed optimization in networks. In *Advances in Neural Information Processing Systems*, pages 2740–2749, 2018.

[50] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.

[51] Daniel A Spielman. Algorithms, graph theory, and linear equations in laplacian matrices. In *Proceedings of the International Congress of Mathematicians 2010 (ICM 2010) (In 4 Volumes) Vol. I: Plenary Lectures and Ceremonies Vols. II–IV: Invited Lectures*, pages 2698–2722. World Scientific, 2010.

[52] Dennis Sundman, Saikat Chatterjee, and Mikael Skoglund. Design and analysis of a greedy pursuit for distributed compressed sensing. *IEEE Transactions on Signal Processing*, 64(11):2803–2818, 2016.

[53] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[54] Joel A Tropp, Anna C Gilbert, and Martin J Strauss. Algorithms for simultaneous sparse approximation. part i: Greedy pursuit. *Signal processing*, 86(3):572–588, 2006.

[55] Berwin A Turlach, William N Venables, and Stephen J Wright. Simultaneous variable selection. *Technometrics*, 47(3):349–363, 2005.

[56] Paul Vanhaesebrouck, Aurélien Bellet, and Marc Tommasi. Decentralized collaborative learning of personalized models over networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[57] Nisheeth K Vishnoi. Laplacian solvers and their algorithmic applications. *Theoretical Computer Science*, 8(1-2):1–141, 2012.

[58] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

[59] Jialei Wang, Mladen Kolar, and Nathan Srerbo. Distributed multi-task learning. In *Artificial Intelligence and Statistics*, pages 751–760, 2016.

[60] Weiran Wang, Jialei Wang, Mladen Kolar, and Nathan Srebro. Distributed stochastic multi-task learning with graph regularization. *arXiv preprint arXiv:1802.03830*, 02 2018.

[61] Yu-Xiang Wang, James Sharpnack, Alexander J Smola, and Ryan J Tibshirani. Trend filtering on graphs. *The Journal of Machine Learning Research*, 17(1):3651–3691, 2016.

[62] Junfeng Yang and Yin Zhang. Alternating direction algorithms for \ell_1-problems in compressive sensing. *SIAM journal on scientific computing*, 33(1):250–278, 2011.

[63] Cun-Hui Zhang, Jian Huang, et al. The sparsity and bias of the lasso selection in high-dimensional linear regression. *The Annals of Statistics*, 36(4):1567–1594, 2008.

[64] Peng Zhao, Guilherme Rocha, Bin Yu, et al. The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics*, 37(6A):3468–3497, 2009.

[65] Jiayu Zhou, Jianhui Chen, and Jieping Ye. Malsar: Multi-task learning via structural regularization. *Arizona State University*, 2011.

A Additional Material - Noiseless setting

In this section we present additional material associated to the noiseless setting. Section A.1 presents simulations results for the solution of the Total Variation Basis Pursuit problem. Section A.2 presents experiments related to the optimisation performance of the distributed algorithm.

A.1 Noiseless Simulations - Sample Complexity

In this section we present simulations associated to the noiseless case for the solution to the Total Variation Basis Pursuit problem (3). To compute the solution we will consider the reformulation into a standard Basis Pursuit Problem (4). Figure 2 then plots the probability of recovery against the number of samples held by non-root nodes N_v for $v \in V \backslash \{1\}$ with a fixed number of root agent samples $N_1 = \lfloor 2s \log(ed/s) \rfloor$. Observe, for a path topology and balanced tree topology, once the non-root nodes have beyond approximately 30 samples, the solution to the reformulated Total Variation Basis Pursuit problem (4) finds the correct support for all of the graph sizes. In contrast, the number of samples required to recover a signal with Basis Pursuit at the same level of sparsity and dimension considered would require at least 80 samples i.e $2s \log(ed/s)$. We therefore save approximately 50 for each non-root problem. We also note as the number of agents n grows, the additional number of samples required to achieve recovery grows more slowly than suggest by Theorem 1, namely, $N_v \gtrsim n^2 s'$.

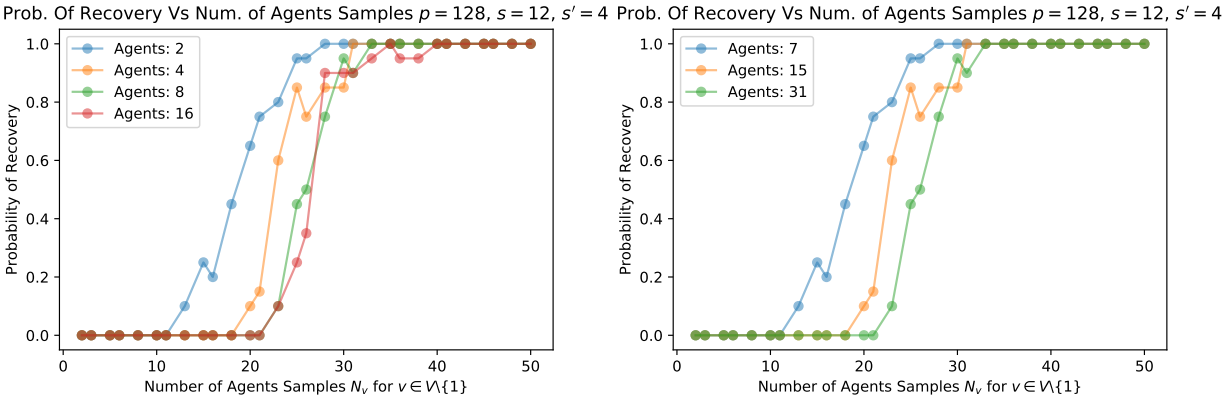


Figure 2: Probability of recovery vs number of non-root node samples N_v for $v \in V \backslash \{1\}$. Problem parameter set to $p = 128$, $s = 12$, $s' = 4$ and $N_1 = \lfloor 2s \log(ed/s) \rfloor = 80$, for path (*Left*) and balance tree with branches of size 2 (*Right*). Each line indicates different size graph, with $n \in \{2, 4, 8, 16\}$ for the path topology and $n \in \{7, 15, 31\}$ for the balanced tree topologies with heights of $\{2, 3, 4\}$ respectively. Solution to reformulated problem (4) found using CVXOPT. Each point is an average of 20 replications. Signal values randomly sampled from $\{1, -1\}$, signal differences are concatenation of s' values. $\{A_v\}$ are standard Gaussian.

A.2 Distributed ADMM Algorithm

In this section present the Distributed ADMM algorithm for solving the Total Variation Basis Pursuit problem. We begin by reformulating the problem into an consensus optimisation form.

Specifically, with $\Delta_e = x_v - x_w$ for $e = \{v, w\} \in E$, we consider

$$\min_{x_v, v \in V} \|x_1\|_1 + \sum_{e \in V} \|\Delta_e\|_1 \text{ subject to}$$
$$A_v x_v = Y_v \text{ for all } v \in V \text{ and } x_v - x_w = \Delta_e \text{ for all } e = \{v, w\} \in E.$$

We now propose the Alternating Direction Method of Multipliers (ADMM) to solve the above. The key step is consider the augmented Lagrangian from dualizing the consensus constraint which, with $\|x\|_1 = \|x_1\|_1 + \sum_{e \in E} \|\Delta_e\|_1$, is

$$\mathcal{L}_\rho(\{x_v\}_{v \in V}, \{\Delta_e\}_{e \in E}, \{\gamma_e\}_{e \in E}) = \|x\|_1 + \sum_{e = \{v,w\} \in E} \frac{\rho}{2} \|x_v - x_w - \Delta_e\|_2 + \langle \gamma_e, x_v - x_w - \Delta_e \rangle.$$

The ADMM algorithm then proceeds to minimise $\mathcal{L}_\rho$ with respect to $\{x_v\}_{v \in V}$, then $\{\Delta_e\}_{e \in E}$, followed by a ascent step in the dual variable $\{\gamma_e\}_{e \in E}$. Full details of the ADMM updates have been given in Appendix C. Each step can be computed in closed form, expect for the update for x_1 which requires solving a basis pursuit problem with an ℓ_2 term in the objective. This can be solved to a high precision efficiently by utilising a simple dual method, see [39, Appendix B]. The additional computational required by the root node in this case aligns with the framework we consider, since we assume the root node also has an additional number of samples N_1.

The theoretical convergence guarantees of ADMM have gained much attention lately due to the wide applicability of ADMM to distributed optimisation problems [5, 22, 24]. While a full investigation of the convergence guarantees of ADMM in this instance is outside the scope of this work, we note for convex objectives with proximal gradient steps computed exactly, ADMM has been shown to converge at worst case a polynomial rate of order $1/t$ [22]. A number of works have shown linear convergence under additional assumptions which include full column rank on the constraints or strong convexity, which are not satisfied in our case [1]. Although, if one considers a proximal variant of ADMM with an additional smoothing term, linear convergence can be shown in the absence of the column rank constraint [24]. The convergence of ADMM can be sensitive hyperparameter choice ρ, which has motivated a number of adaptive schemes, see for instance [23].

We now discuss the empirical optimisation performance for the Distributed ADMM algorithm just introduce. For investigating the optimisation performance, we compute the solution to the reformulated problem using a standard Basis Pursuit solver as in Section A.1, and then compare it to the solution found by the ADMM method. Looking to Figure 3 we see the optimisation error ($\log_{10}$ axis) vs the number of ADMM iterations, for path and balanced tree topologies. The error is seen to converge with a linear rate. The convergence for a path topology is naturally slower, reaching a precision of 10^{-8} in 300 iterations for 7 nodes, while the same size balanced tree topology reaches a precision of 10^{-15}. This is expected as the balanced trees considered are more connected than a path, and therefore, information propagates around the nodes quicker. Larger tree topologies also require additional iterations to reach the same precision, with a size 63 tree reaching a precision of $10^{-7.5}$ in 300 iterations.

[1] The constraint dualised by ADMM, $x_v - x_w = \Delta_e$ for $e = \{v, w\} \in E$, can be denoted in terms the signed incident matrix of the graph. This is a linear constraint, but the signed incident matrix does not have full column rank.

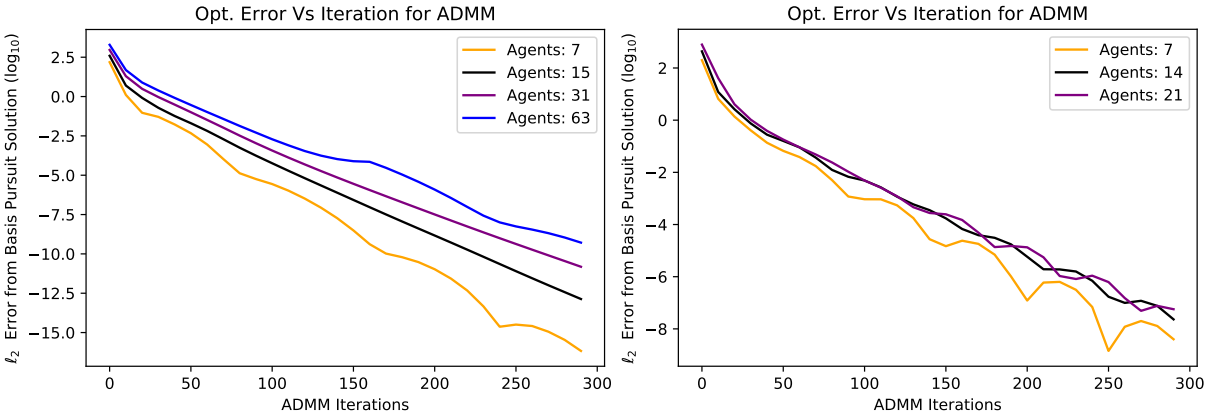


Figure 3: Optimisation error $\|x^t - x^\star_{\text{BP}}\|_2^2$ (Log scale) vs Iterations for ADMM method (10) with $\rho = 10$ for different graph sizes (lines) and topologies (plots). Here $x^\star_{\text{BP}}$ is the Basis Pursuit solution to the reformulated problem (4) using *CVXOPT*. Problem parameters $p = 2^9$, $s = \lfloor 0.1p \rfloor$ and $s' = 4$. *Left*: Balanced trees, branch size 2 and heights $\{2, 3, 4, 5\}$. *Right*: Path topology. Agent sample size $N_1 = 2s \log(ep/2s)$ and $N_v = 150$ for $v \neq 1$. Matrices $\{A_v\}_{v \in V}$ i.i.d standard Gaussian entries, $x_1^\star$ has s values randomly drawn from $\{+1, -1\}$ and $\{\Delta_e^\star\}_{e \in E}$ each have s' i.i.d standard Gaussian entries, locations chosen at random.

B Additional Material - Noisy Setting

In this section we present additional material associated to the noisy setting within the main body of the manuscript. Section B.1 presents plots for experiments on simulated data. Section B.2 presents plots for experiments with real data.

B.1 Additional Plots for Total Variation Basis Pursuit Denoising - Simulated Data

In this section we present experiments comparing the performance of Total Variation Basis Pursuit Denoising (9) to the group lasso and dirty model of [28]. Figure 5 plots the ℓ_1 estimation error for each of these methods with simulated data in the case of path and balanced tree topologies. Observe, as the number of agents increases, that the estimation error for the group lasso and dirty model grows quicker than for Total Variation Basis Pursuit Denoising. In the case of a path topology, this is particularly noticeable and is because the size of support for the agent furthest from the root increases with the number of agents. Meanwhile for balanced trees considered, the size of support for the agent furthest from the root remains fixed as the number of agents grows (the tree height is fixed). The flexibility of the dirty model to fit a node specific component in this case allows it account for variation in support across tasks, and thus, scale more gracefully than the group lasso as the number of agents grows. Although, the estimation error remains noticeably higher than the total variation basis pursuit approach. This is due to the dirty model estimating a specialised component for each task, while the Total Variation encodes tree structure.

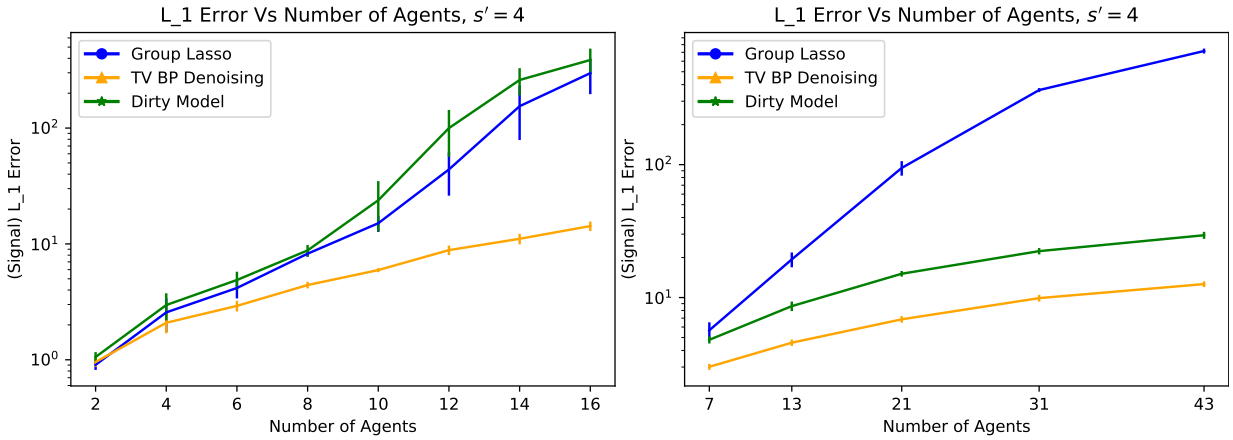


Figure 4: ℓ_1 estimation error $\sum_{v \in V} \|x_v - x_v^\star\|_1$ ($\log_{10}$ scale) against number of agents for Total Variation Basis Pursuit Denoising solved using SPGL1 Python package (*Yellow*), Group Lasso (*blue*) and dirty model of [28] (*Green*). *Left*: Path topology. *Right*: Balanced tree topology height 2 branching rate $\{2, 3, 4, 5, 6\}$. The same i.i.d standard Gaussian matrix was associated to each node with $N_v = 200$ for $v \in V$, and other problem parameters were $p = 2^9$, $s = 25$ and $s' = 4$. Signal at the root $x_1^\star$ and the differences $\{\Delta_e^\star\}_{e \in E}$ random sampled from $\{+1, -1\}$, with no overlap in supports i.e. as described at end of Section 2.2. Group lasso used best regularisation from between $[10^{-6}, 10^{-2}]$. Dirty model regularisation followed [28] with (in their notation) 5×5 (log -scale) grid search for λ_g and λ_b with $\lambda_g/\lambda_b \in [10^{-3}, 10]$, $\lambda_b = c\sqrt{7/200}$ and $c \in [10^{-2}, 10]$. Dirty model was fit using MALSAR [65]. The group lasso variants used normalised matrices $A_v/\sqrt{N_v}$ and responses $y_v/\sqrt{N_v}$. Total Variation Basis Pursuit Denoising parameter was $\eta = \sqrt{200 \times n}0.1$. Each point and error bars from 5 replications. Identical plot with natural axis in Figure 5 Appendix B.1.

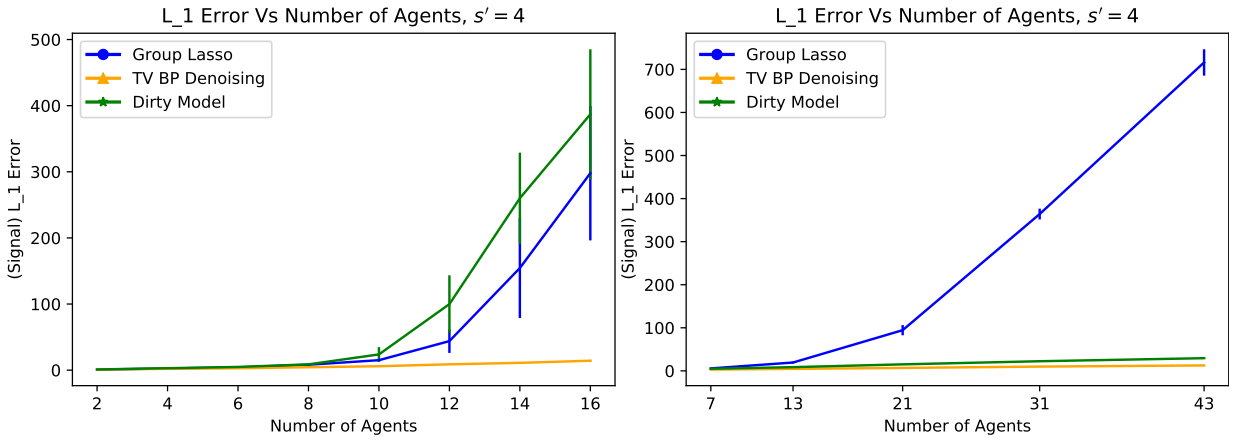


Figure 5: Identical plot to Figure 4, but with natural y-axis.

B.2 Data Preparation and Experiment Parameters for AVIRIS Application

In this section we present the application of Total Variation Basis Pursuit Denoising to the AVIRIS Cuprite dataset. We begin with Figure 6, which presents the sector of the AVIRIS Cuprite dataset used, as well as the 80 x 80 pixel subset portion sub-sampled for our experiment. We note each pixel in the dataset is associated to 224 spectral bands between 400 and 2500 nm and, in short, the objective is to decompose the spectrum of each pixel into a sparse linear combination known mineral spectra. The specific bandwidth presented in Figure 6 demonstrate that this area maybe a region of interest. Following [26, 27], we construct a spectral library A_{Lib} by randomly sampling 240 mineral from the USGS library splib07 [2]. After cleaning the AVIRIS dataset and the library we are left with $N_v = 184$ spectral bands for each pixel $v \in V$, and thus, $A_v = A_{\text{Lib}} \in \mathbb{R}^{184\times 240}$ and $y_v \in \mathbb{R}^{184}$. We now go on to describe more detail the experimental steps.

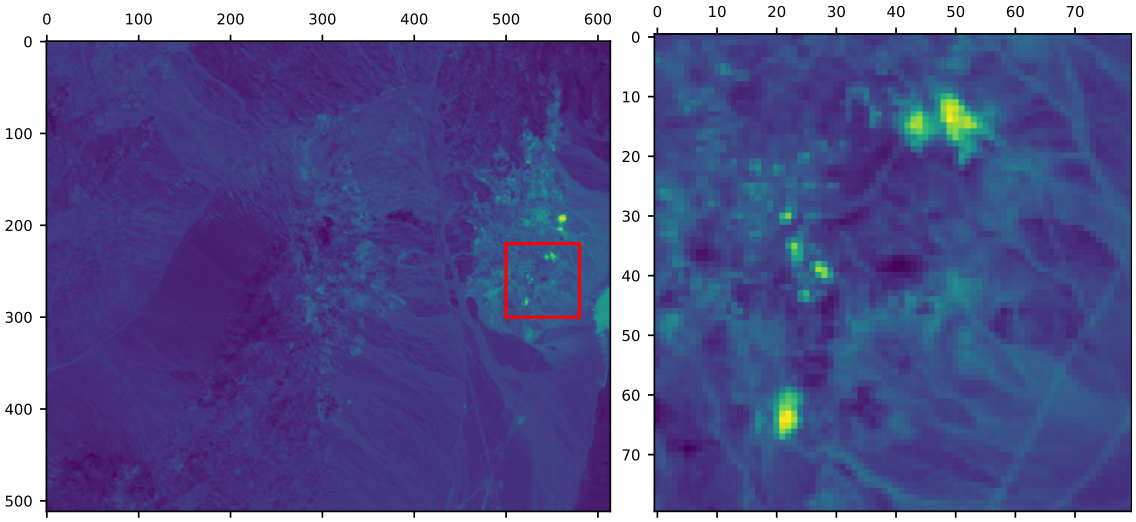


Figure 6: *Left*: Sector `f970619t01p02_r04_sc03.a.rfl` of AVIRIS data set at bandwidth of 557.07 nm. Red square indicates 80×80 portion of the sector used as the data set. *Right*: Red squared section zoomed in.

Cleaning AVIRIS Cuprite Dataset We followed [27] and removed the spectral bands 1-2, 105-115, 150-170 and 223-224, which are due to water absorption and low signal to noise. This would leave us with 188 spectral bands, although additional bands were removed due to large values within the USGS Library, see next paragraph.

Sub-sampling USGS Library We took a random sample of 240 minerals from splib07 library, that are specifically calibrated to the AVIRIS 1997 data set i.e. have been resampled at the appropriate bandwidths. A number of the spectrum for the minerals were corrupted or had large reflectance values for particular wavelengths e.g. greater than 10^{34}. We therefore restricted ourselves to minerals that had less than 10 corrupted wavelengths. After sub-sampling, any wavelengths with a corrupted value (if it contained a value greater than 10) were removed. This left us with 184 spectral bands.

[2] `https://crustal.usgs.gov/speclab/QueryAll07a.php`

Algorithm Parameters To apply Basis Pursuit Denoising independently to each pixel, we used the SPGL1 python package, which can be found at `https://pypi.org/project/spgl1/`. To solve the Total Variation Basis Pursuit Denoising problem (9), we used the Alternating Direction Methods of Multiplers (ADMM) algorithm for ℓ_1-problems in [62], specifically the inexact method (2.16). We applied this algorithm to the normalised data i.e. dividing by the matrix and response vector by the square root of the total number of samples (4 pixels $\times$ 184 spectral bands). We ran the algorithm for 500 iterations with parameters (in the notation of [62]) $\tau = 0.1$, $\beta = 2$, $\gamma = 0.1$ and $\delta = 0.001$. We note that directly applying the SPGL1 python package to the Total Variation Basis Pursuit Denoising problem (9), resulted in instabilities when choosing $\eta < 0.2$. We chose $\eta = 0.001$ for both independent Basis Pursuit Denoising case and the Total Variation Basis Pursuit Denoising (9), following the regularisation choice in [27]. Meanwhile, the group lasso was fit using scikit-learn with regularisation 0.001, and the SUNnSAL algorithm [3] with regularisation 0.001 was applied using the python implementation which can be found at `https://github.com/Laadr/SUNSAL`. We note when using SUNnSAL it is common to perform a computationally expensive pre-processing step involving a non-convex objective, see [26, 27]. This was not performed in this case, as all of the other methods did not pre-process the data.

C Distributed ADMM Updates for Total Variation Basis Pursuit

In this section we more precisely describe the Distributed ADMM algorithm for fitting the Total Variation Basis Pursuit problem (4). We recall the consensus optimisation formulation of the Total Variation Basis Pursuit problem is as follows

$$\begin{aligned}&\min_{x_v, v\in V} \|x_1\|_1 + \sum_{e\in V} \|\Delta_e\|_1 \text{ subject to}\\ &A_v x_v = Y_v \text{ for all } v \in V\\ &x_v - x_w = \Delta_e \text{ for all } e = \{v, w\} \in E.\end{aligned}$$

where we consider the Augmented Lagrangian from dualizing the consensus constraint

$$\begin{aligned}&\mathcal{L}_\rho(\{x_v\}_{v\in V}, \{\Delta_e\}_{e\in E}, \{\gamma_e\}_{e\in E}) = \|x_1\|_1\\ &+ \sum_{e=\{v,w\}\in E} \|\Delta_e\|_1 + \frac{\rho}{2}\|x_v - x_w - \Delta_e\|_2 + \langle \gamma_e, x_v - x_w - \Delta_e\rangle.\end{aligned}$$

Now the ADMM algorithm initialized at $(\{x_v^1\}_{v\in V}, \{\Delta_e^1\}_{e\in E}, \{\gamma_e^1\}_{e\in E})$ then proceeds to update the iterates for $t \geq 1$ as

$$\begin{aligned}x_v^{t+1} &= \arg\min_{x_v^t} \mathcal{L}_\rho(\{x_v^t\}_{v\in V}, \{\Delta_e^t\}_{e\in E}, \{\gamma_e^t\}_{e\in E}) \text{ subject to } A_v x_v = Y_v \text{ for } v \in V && (10)\\ \Delta_e^{t+1} &= \arg\min_{\Delta_v^t} \mathcal{L}_\rho(\{x_v^{t+1}\}_{v\in V}, \{\Delta_e^t\}_{e\in E}, \{\gamma_e^t\}_{e\in E}) && \text{for } e \in E\\ \gamma_e^{t+1} &= \gamma_e^t + \rho(x_v - x_w - \Delta_e) && \text{for } e \in E\end{aligned}$$

We now set to show how each of the above updates can be implemented in a manner that respects the network topology due to the Augmented Lagrangian $\mathcal{L}_\rho$ decoupling across the network. These will be precisely described within the following sections. For clarity each update will be given its own subsection and the super script notation i.e. x_v^t will be suppressed.

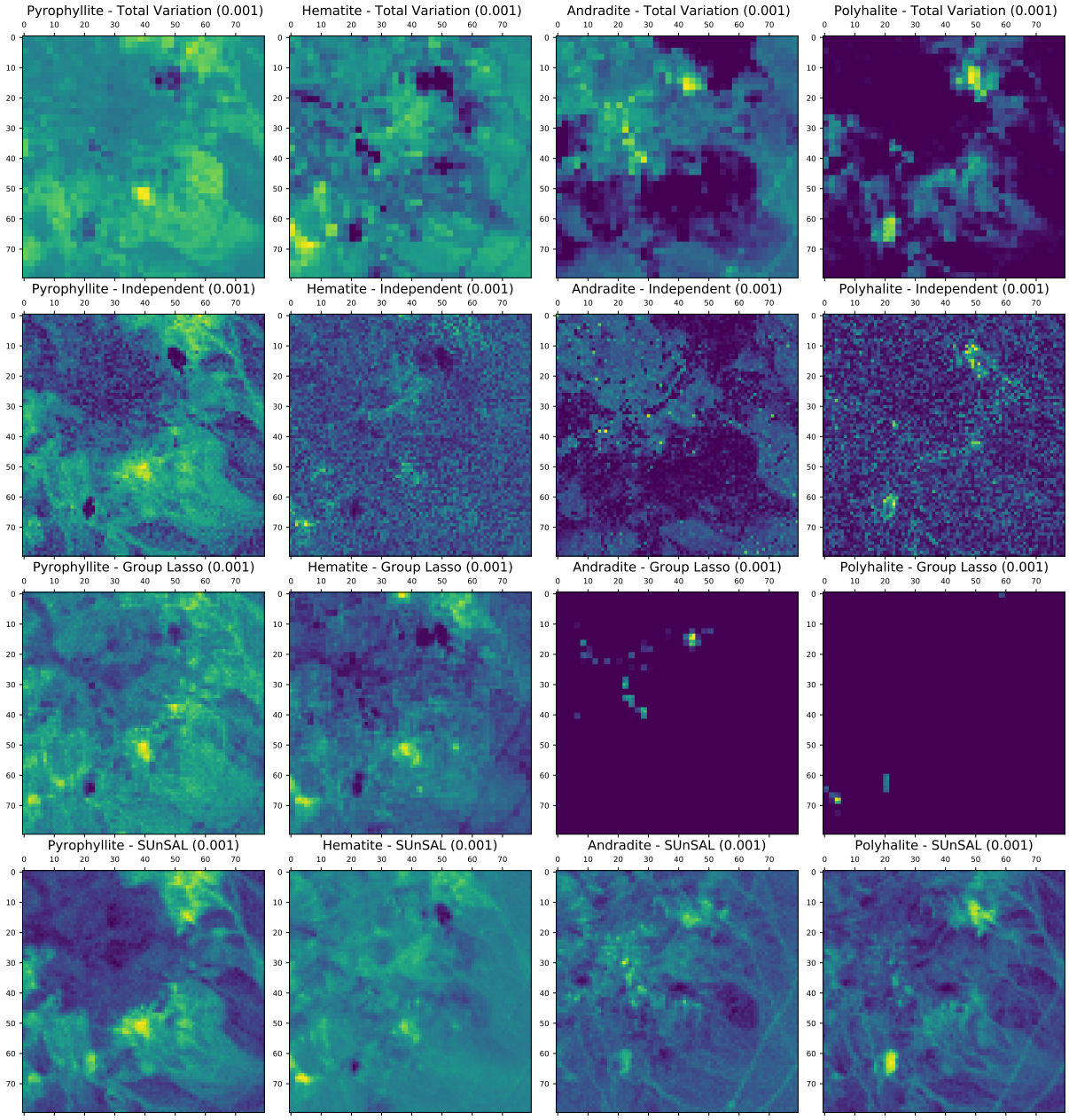


Figure 7: Coefficients associated to the mineral Pyrophyllite (*Left*), Hematite (*Left-Middle*), Andradite (*right-middle*), and Polyhalite (*Right*). Methods considered are: *Top*: Total Variation Basis Pursuit Denoising applied to 2x2 pixels simultaneously with $\eta = 0.001$; *Middle-Top*: Basis Pursuit Denoising applied independently to each pixel with $\eta = 0.001$. *Middle-Bottom*: group lasso (jointly penalised all coefficients) applied to 2x2 pixels simultaneously with regularisation 0.001. *Bottom*: SUNSAL with regularisation of 0.001. Yellow pixels indicate higher values.

C.1 Updating $\{x_v\}$

The updates for $\{x_v\}_{v\in V}$ take two different forms depending on whether v is associated to the root node i.e. $v = 1$ or otherwise. We begin with the case of a root note.

C.1.1 Root Node x_1

The update for x_1 in the ADMM algorithm (10) requires solving

$$\begin{aligned}\min_{x_1} \|x_1\|_1 + &\sum_{e=(i,j)\in E:i=1} \frac{\rho}{2}\|x_1 - x_j - \Delta_e\|_2^2 + \langle \gamma_e, x_1 - x_j - \Delta_e\rangle \\ &+ \sum_{e=(i,j)\in E:j=1} \frac{\rho}{2}\|x_j - x_1 - \Delta_e\|_2^2 + \langle \gamma_e, x_j - x_1 - \Delta_e\rangle \\ \text{subject to } & A_1 x_1 = y_1\end{aligned}$$

where we note the two summations in the objective arise from the orientation of the edges within the network. This is then equivalent to considering solve a problem of the form

$$\min_x \|x\|_1 + \nu^\top x + c\|x\|_2^2 \text{ subject to } Ax = b \tag{11}$$

with parameters $A = A_1$, $b = y_1$, $c = \text{Deg}(1)\frac{\rho}{2}$ where $\text{Deg}(1)$ is the degree of the root node 1 and $\nu = \sum_{e=(i,j)\in E:i=1} \gamma_e - \rho(x_i + \Delta_e) + \sum_{e=(i,j)\in E:j=1} -\gamma_e - \rho(x_j + \Delta_e)$.

To solve the problem (11) we adopt the approach used in [39, Appendix B] to an optimisation problem of the same form. That is, we consider the dual problem

$$\max_\lambda \lambda^\top b + \sum_{i=1}^p \inf_{x_i} \left(|x_i| + u_i(\lambda)x_i + cx_i^2\right)$$

where the dual variable $\lambda \in \mathbb{R}^n$ and $u(\lambda) = \nu - A^\top\lambda$. The gradient of the above problem is then $b - Ax(\lambda)$ where $x(\lambda) = (x(\lambda)_1, \ldots, x(\lambda)_p)$ is constructed from the unique minimiser of $|x_i| + u_i(\lambda)x_i + cx_i^2$ for $i = 1, \ldots, p$ which is $x(\lambda)_i$. This can then be written in closed form as

$$x_i(\lambda) = \begin{cases} 0 & \text{if } -1 \le u_i(\lambda) \le 1 \\ -(u_i(\lambda) + 1)/2c & \text{if } u_i(\lambda) < -1 \\ -(u_i(\lambda) - 1)/2c & \text{if } u_i(\lambda) > 1 \end{cases}$$

Given a solution $\lambda^\star$ the solution to the original problem is then $x(\lambda^\star)$. To solve the Dual problem we use the Barzilai - Borwein algorithm [46] with warm restarts using the dual variable from the previous iteration.

C.1.2 Non-Root Node

In the case of x_v which is not the root node i.e. $v \neq 1$, we require solving the optimisation problem

$$\begin{aligned}\min_{x_v} &\sum_{e=(i,j)\in E:i=v} \frac{\rho}{2}\|x_v - x_j - \Delta_e\|_2^2 + \langle \gamma_e, x_v - x_j - \Delta_e\rangle \\ &+ \sum_{e=(i,j)\in E:j=v} \frac{\rho}{2}\|x_i - x_v - \Delta_e\|_2^2 + \langle \gamma_e, x_i - x_v - \Delta_e\rangle \\ \text{subject to } & A_v x_v = y_v\end{aligned}$$

This minimisation can be written in the form

$$\min_x \|x\|_2^2 + \langle a, x\rangle \text{ subject to} \quad (12)$$
$$Ax = b$$

with parameters $A = A_v$, $b = y_v$ and
$a = \frac{2}{\mathrm{Deg}(v)}\Big(\big(\sum_{e\in\{i,j\}:i=v} -\Delta_e - x_j + \frac{\gamma_e}{\rho}\big) + \big(\sum_{e\in\{i,j\}:j=v} \Delta_e - x_i - \frac{\gamma_e}{\rho}\big)\Big)$. Since $\|x\|_2^2 + \langle a, x\rangle = \|x + \frac{a}{2}\|_2^2 - \frac{1}{2}\|a\|_2^2$, This leads to the equivalent optimisation problem

$$\min_u \|u\|_2^2 \text{ subject to}$$
$$Au = b + A\frac{a}{2}.$$

This is exactly the least norm solution to a linear system, and is solved by $u = A^\dagger(b + A\frac{a}{2})$ where $A^\dagger$ is the Moore-Penrose pseudo-inverse. We then recover the solution to (12) by setting $x = A^\dagger(b + A\frac{a}{2}) - \frac{a}{2}$.

C.2 Updating $\{\Delta_e\}_{e\in V}$

For each edge $e = (i, j) \in E$ the updates require solving

$$\min_{\Delta_e} \|\Delta_e\|_1 + \frac{\rho}{2}\|x_i - x_j - \Delta_e\|_2^2 - \langle\gamma_e, \Delta_e\rangle$$

which is a equivalent to

$$\min_{\Delta_e} \|\Delta_e\|_1 + \frac{\rho}{2}\|\Delta_e\|_2^2 - \langle\Delta_e, \gamma_e + z_i - z_j\rangle.$$

This is a shrinkage step and thus the minimiser can be written as

$$\Delta_e = \begin{cases} 0 & \text{if } |\gamma_e + \rho(z_i - z_j)| < 1 \\ \frac{1}{\rho}(\gamma_e + \rho(z_i - z_j) - 1) & \text{if } \gamma_e + \rho(z_i - z_j) > 1 \\ \frac{1}{\rho}(\gamma_e + \rho(z_i - z_j) + 1) & \text{if } \gamma_e + \rho(z_i - z_j) < -1 \end{cases}$$

D Concentration Theorem for RIP and Proof of Lemma 1

We begin with Theorem 9.2 from [21], which demonstrates that a sub-Gaussian matrix can satisfy the Restricted Isometry Property in high probability provided the sample size is sufficiently large.

Theorem 3. *Let $A \in \mathbb{R}^{N\times p}$ be sub-Gaussian matrix with independent and identically distributed entries. Then there exists a constant $C > 0$ (depending on sub-Gaussian parameters β and κ) such that the Restricted Isometry Constant of $A/\sqrt{N}$ satisfies $c_k \le \delta$ with probability atleast $1-\epsilon$ provided*

$$N \ge C\delta^{-2}(k\log(ep/k) + \log(\epsilon/2)).$$

We now proceed to the provide the proof of Lemma 1. We begin with the following proposition from Proposition 6.3 in [21].

Proposition 1. *Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^p$ be vectors such that $\|\mathbf{u}\|_0 \leq s$ and $\|\mathbf{v}\|_0 \leq k$, and matrix $A \in \mathbb{R}^{N \times p}$ satisfy Restricted Isometry Property up to $s+k$ with constant c_{s+k}. If the support of the vectors is disjoint $Supp(v) \cap Supp(u) = \emptyset$ then*

$$|\langle A\mathbf{u}, A\mathbf{v}\rangle| \leq c_{s+k}\|\mathbf{u}\|\|\mathbf{v}\|_2$$

We now begin the proof of Lemma 1.

Lemma 1. Recall we have $x = (x_1, \Delta_1, \ldots, \Delta_{|E|}) \in \text{Ker}(A)\backslash\{0\}$ and as such for each $v \in V$ we have the condition

$$A_v(x_1 + \sum_{e \in \pi(v)} \Delta_e) = 0$$

We now begin by proving the inequality (5), which is an upper bound on $\|(x_1 + \sum_{e\in\pi(v)} \Delta_e)_U\|_1$ for subsets $U \subseteq \{1, \ldots, p\}$ such that $|U| \leq s'$. Note is suffices to consider the case where U is the largest s' indexes of the vector $x_1 + \sum_{e\in\pi(v)} \Delta_e$. To lower notational burden, we will simply denote $x_{\pi(v)} = x_1 + \sum_{e\in\pi(v)} \Delta_e$. Now, from the above equality we have $A_v(x_{\pi(v)})_U = -A_v(x_{\pi(v)})_{U^c}$ and thus

$$(1 - \delta_{s'})\|(x_{\pi(v)})_U\|_2^2 \leq \|A(x_{\pi(v)})_U\|_2^2 = -\langle A(x_{\pi(v)})_U, A(x_{\pi(v)})_{U^c}\rangle.$$

where the first inequality arises from the Restricted Isometry Property of A_v at sparsity level s'. Now, proceed to decompose U^c into disjoint sets each of size s'. In particular let $U^c = B_1 \cup B_2 \cup B_3 \cup \ldots$ such that $B_j \cap B_i = \emptyset$ for $i \neq j$, and $|B_j| \leq s'$ for all $j = 1, 2, \ldots$. The sets in this decomposition are then defined recursively. Specifically, let B_1 be the indexes of the largest s' entries of $x_{\pi(v)}$ restricted to the indices in U^c. Similarly, let B_2 be the indexes of the largest s' entries of $x_{\pi(v)}$ restricted to the indices in $(U \cup B_1)^c$. More generally, for $j = 3, 4, \ldots$ we then let B_j be the indexes of the largest s' entries of $x_{\pi(v)}$ restricted to the indices in in $(U \cup B_1 \cup B_2 \cup \ldots B_{j-1})^c$. This then leads, with Proposition 1 since the sets are disjoint from U, the upper bound

$$\begin{aligned}(1 - \delta_{s'})\|(x_{\pi(v)})_U\|_2^2 &\leq \Big|\sum_{j\geq 1}\langle A(x_{\pi(v)})_U, A(x_{\pi(v)})_{B_j}\rangle\Big| \\ &\leq \delta_{2s'}\|(x_{\pi(v)})_U\|_2 \sum_{j\geq 1}\|(x_{\pi(v)})_{B_j}\|_2.\end{aligned}$$

It now suffices to upper bound $\sum_{j\geq 1}\|(x_{\pi(v)})_{B_j}\|_2$. Note that we then have for $j = 2, \ldots$ the upper bound $\|(x_{\pi(v)})_{B_j}\|_2 \leq \sqrt{s'}\|(x_{\pi(v)})_{B_j}\|_\infty \leq \frac{1}{\sqrt{s'}}\|(x_{\pi(v)})_{B_{j-1}}\|_1$. While we also have for $j = 1$ the upper bound $\|(x_{\pi(v)})_{B_1}\|_2 \leq \frac{1}{\sqrt{s'}}\|(x_{\pi(v)})_U\|_1$. Plugging these bounds into the above, summing up so $\|(x_{\pi(v)})_U\|_1 + \sum_{j\geq 2}\|(x_{\pi(v)})_{B_{j-1}}\|_1 = \|x_{\pi(v)}\|_1$, and dividing both sides by $(1 - \delta_{s'})\|(x_{\pi(v)})_U\|_2$ then yields

$$\|(x_{\pi(v)})_U\|_2 \leq \frac{1}{\sqrt{s'}}\frac{\delta_{2s'}}{1 - \delta_{s'}}\|x_{\pi(v)}\|_1.$$

The inequality (5),is then arrived at by using the lower bound $\|(x_{\pi(v)})_U\|_2 \geq \frac{1}{\sqrt{s'}}\|(x_{\pi(v)})_U\|_1$ as well as that $\delta_{2s'} \geq \delta_{s'}$.

□

E Proof of Theorem 2

In this section we provide a proof of Theorem 2.

Theorem 2. We now set to show that the Robust Null Space Property (8) holds for some ρ, τ. We note it suffices to show the following which is equivalent to the Robust Null Space Property

$$\|(x)_S\|_1 \leq \rho' \|x\|_1 + \tau' \|Ax\|_2 \text{ for all } x \in \mathbb{R}^N.$$

In particular, by adding $\rho\|(x)_S\|_1$ to both sides of the inequality for the Robust Null Space Property (8) and dividing by $1+\rho$, we see that if the above holds then the Robust Null Space Property holds with $\rho = \frac{\rho'}{1-\rho'}$ and $\tau = \tau'/(1-\rho')$.

We begin by closely following the proof of Lemma 1 to control the ℓ_1 norm of $x_1 + \sum_{e\in\pi(v)} \Delta_e = x_{\pi(v)}$ for $v \in V$ restricted to subsets U. Similar to that proof, start by considering subsets U of size $|U| \leq s'$, and in particular, the set U associated to the largest s' entries. Recall the decomposition of $U^c = B_1 \cup B_2 \cup \dots$. Now, observe that we can upper bound

$$\begin{aligned}
(1-\delta_{s'})\|(x_{\pi(v)})_U\|_2^2 &\leq \|A_v(x_{\pi(v)})_U\|_2^2 \\
&= \Big\langle A_v(x_{\pi(v)})_U, A_v\Big(x_{\pi(v)} - \sum_{j\geq 1}(x_{\pi(v)})_{B_j}\Big)\Big\rangle \\
&= \langle A_v(x_{\pi(v)})_U, A_v x_{\pi(v)}\rangle - \sum_{j\geq 1}\langle A_v(x_{\pi(v)})_U, A_v(x_{\pi(v)})_{B_j}\rangle \\
&\leq \sqrt{1+\delta_{s'}}\|(x_{\pi(v)})_U\|_2\|A_v x_{\pi(v)}\|_2 + \frac{\delta_{2s'}}{\sqrt{s'}}\|(x_{\pi(v)})_U\|_2\|x_{\pi(v)}\|_1
\end{aligned}$$

where we simply upper bounded using the Restricted Isometry Property of A_v the inner product $\langle A_v(x_{\pi(v)})_U, A_v x_{\pi(v)}\rangle \leq \|A_v(x_{\pi(v)})_U\|_2\|A_v x_{\pi(v)}\|_2 \leq \sqrt{1+\delta_{s'}}\|(x_{\pi(v)})_U\|_2\|\|A_v x_{\pi(v)}\|_2$ and followed the steps in the proof of Lemma 1 to upper bound $\sum_{j\geq 1}\langle A_v(x_{\pi(v)})_U, A_v(x_{\pi(v)})_{B_j}\rangle \leq \delta_{2s'}\|(x_{\pi(v)})_U\|_2 \sum_{j\geq 1}\|(x_{\pi(v)})_{B_j}\|_1 \leq \frac{1}{\sqrt{s'}}\delta_{2s'}\|(x_{\pi(v)})_U\|_2\|x_{\pi(v)}\|_1$. Dividing both sides by $(1-\delta_{s'})\|(x_{\pi(v)})_U\|_2$ we then get

$$\|(x_{\pi(v)})_U\|_2 \leq \frac{\delta_{2s'}}{1-\delta_{s'}}\frac{1}{\sqrt{s'}}\|x_{\pi(v)}\|_1 + \frac{\sqrt{1+\delta_{s'}}}{1-\delta_{s'}}\|A_v x_{\pi(v)}\|_2$$

Using that $\|(x_{\pi(v)})_U\|_2 \geq \frac{1}{\sqrt{s'}}\|(x_{\pi(v)})_U\|_1$ as well as simply upper bounding $\|x_{\pi(v)}\|_1 = \|x_1 + \sum_{e\in\pi(v)}\Delta_e\|_1 \leq \|x_1\|_1 + \sum_{e\in\pi(v)}\|\Delta_e\|_1 \leq \|x_1\|_1 + \sum_{e\in E}\|\Delta_e\|_1$ we have in a similar manner to Lemma 1

$$\|(x_1 + \sum_{e\in\pi(v)}\Delta_e)_U\|_1 \leq \frac{\delta_{2s'}}{1-\delta_{s'}}(\|x_1\|_1 + \sum_{e\in E}\|\Delta_e\|_1) + \frac{\sqrt{1+\delta_{s'}}}{1-\delta_{s'}}\sqrt{s'}\|A_v x_{\pi(v)}\|_2. \tag{13}$$

For $e = \{v, w\} \in E$ we now set to bound $\|(\Delta_e)_{S_e}\|_1$ where recall S_e are the largest s' elements of Δ_e. Following the proof of Theorem 1, suppose w is closest to the root node. If not, swap the v, w in the following. By adding and subtracting $\big(x_1 + \sum_{\widetilde{e}\in\pi(w)}\Delta_{\widetilde{e}}\big)_{S_e}$ we then get

$$\begin{aligned}
\|(\Delta_e)_{S_e}\|_1 &\leq \|(x_1 + \sum_{\widetilde{e}\in\pi(w)}\Delta_{\widetilde{e}})_{S_e}\|_1 + \|(x_1 + \sum_{\widetilde{e}\in\pi(v)}\Delta_{\widetilde{e}})_{S_e}\|_1 \\
&\leq \frac{2\delta_{2s'}}{1-\delta_{s'}}(\|x_1\|_1 + \sum_{e\in E}\|\Delta_e\|_1) + \frac{\sqrt{1+\delta_{s'}}}{1-\delta_{s'}}\sqrt{s'}(\|A_v x_{\pi(v)}\|_2 + \|A_w x_{\pi(w)}\|_2)
\end{aligned}$$

where on the second inequality we applied (13) twice. Summing the above over all edges $e \in E$, we note $\|A_v x_{\pi(v)}\|_2$ for $v \in V$ appears at most the max degree of the graph, as such we get

$$\begin{aligned}\sum_{e\in E} \|(\Delta_e)_{S_e}\|_1 &\leq \frac{2N\delta_{2s'}}{1-\delta_{s'}}(\|x_1\|_1 + \sum_{e\in E}\|\Delta_e\|_1) + \frac{\sqrt{1+\delta_{s'}}}{1-\delta_{s'}}\mathrm{Deg}(G)\sqrt{s'}\sum_{v\in V}\|A_v x_{\pi(v)}\|_2 \\ &\leq \frac{2N\delta_{2s'}}{1-\delta_{s'}}(\|x_1\|_1 + \sum_{e\in E}\|\Delta_e\|_1) + \frac{\sqrt{1+\delta_{s'}}}{1-\delta_{s'}}\mathrm{Deg}(G)\sqrt{ns'}\sqrt{\sum_{v\in V}\|A_v x_{\pi(v)}\|_2^2}\end{aligned}$$

where on the final inequality we upper bounded $\sum_{v\in V}\|A_v x_{\pi(v)}\|_2 \leq \sqrt{n}\sqrt{\sum_{v\in V}\|A_v x_{\pi(v)}\|_2^2}$.

We now consider the bound for $\|(x_1)_U\|_1$ but for subsets U of size up to s. Following an identical set of steps as for (13), but with s' swapped with s and $\delta_{s'}$ swapped with $\delta_s^{(1)}$, we get the upper bound

$$\begin{aligned}\|(x_1)_U\|_1 &\leq \frac{\delta_{2s}^{(1)}}{1-\delta_s^{(1)}}(\|x_1\|_1 + \sum_{e\in E}\|\Delta_e\|_1) + \frac{\sqrt{1+\delta_s^{(1)}}}{1-\delta_s^{(1)}}\sqrt{s}\|A_1 x_1\|_2 \\ &\leq \frac{\delta_{2s}^{(1)}}{1-\delta_s^{(1)}}(\|x_1\|_1 + \sum_{e\in E}\|\Delta_e\|_1) + \frac{\sqrt{1+\delta_s^{(1)}}}{1-\delta_s^{(1)}}\sqrt{s}\sum_{v\in V}\sqrt{\|A_v x_v\|_2^2}\end{aligned}$$

where at the end we simply upper bounded $\|A_1 x_1\|_2 = \sqrt{\|A_1 x_1\|_2^2} \leq \sqrt{\sum_{v\in V}\|A_v x_v\|_2^2}$. Picking $U = S_1$, and adding together the upper bound for $\sum_{e\in E}\|(\Delta_e)_{S_e}\|_1$ and $\|(x_1)_U\|_1$ and collecting terms then yields the result. □

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

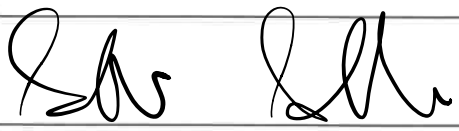
Title of Paper	Tree-Based Multi-Task Sparse Recovery with Total Variation Penalty
Publication Status	☐Submitted for Publication
Publication Details	"Tree-Based Multi-Task Sparse Recovery with Total Variation Penalty'', Dominic Richards, Sahand Negahban, Patrick Rebeschini. In Preprint, 2020

Student Confirmation

Student Name:	Dominic Richards		
Contribution to the Paper	Formulated main idea alongside Sahand Negahban. Later refined the idea alongside Patrick Rebeschini. Derived main technical results and coded experiments. Wrote first draft of manuscript, later versions written alongside Patrick Rebeschini.		
Signature		Date	04/01/2021.

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	PATRICK REBESCHINI ASSOCIATE PROFESSOR		
Supervisor comments			
Signature		Date	11/01/2021

This completed form should be included in the thesis, at the end of the relevant chapter.

6
Conclusion

The work in this thesis has investigated approaches for utilising statistics in the framework of distributed machine learning. In the second, third and fourth chapters, associated to the works [35–37], we considered a *homogeneous* setting where agents in a decentralised network have datasets sampled independently and identically from the same population. Given this setting and motivated by the successes of iterative first order methods in artificial intelligence, we investigated the learning performance of a simple iterative decentralised algorithm: Distributed Gradient Descent [38]. In the case of non-parametric regression, we found that the statistical setting can be leveraged, namely, the statistical concentration of quantities held by the agents, to achieve a speed-up in computational performance for *any* network topology. This being in contrast to prior work which studies the problem through the more general lens of consensus optimisation, and thus, results in a slow-down in computational performance for poorly connected network topologies [29, 30]. Meanwhile, for more general loss functions it was found that the implicit regularisation effects of gradient descent can be extended to the decentralised setting, and thus, allow for simple algorithms to achieve generalisation guarantees without the need for constraints and explicit regularisation.

The aforementioned works motivate a number of possible future research directions in the homogeneous setting, which we now briefly discuss within the

following four paragraphs.

Linear Speed-up for General Losses We note that the theoretically guaranteed linear speed-up in computational run time is currently limited to the setting of non-parametric regression. Extending this result to more general losses will require generalising techniques currently to the specialised case for squared loss with linear models. Precisely, the case of non-parametric regression allows fine-grained control on the deviation between the agents iterates and the entire network average i.e. *decentralised error*. This is for two reasons. Firstly, the gradients are linear in the parameters allowing the network average to be closely related to the equivalent single-machine algorithm (Distributed Gradient Descent on a complete graph topology). Secondly, the contraction of the gradient updates towards a minimiser is explicit, allowing for more refined bounds to be achieved. Extending these observations to a general loss is challenging as the equivalent single machine algorithm (run on a complete graph) is not as closely related to the entire network average as the gradients can be non-linear.

Different Sample Sizes Across the Network Currently Chapters 2, 3 and 4 assume the sample size at each agent is the same, leaving open the question of what occurs when agents have different sample sizes. One challenge in this direction arises from the requirement that *every agent* achieves the optimal statistical rate with respect to all of the samples in the network. Specifically, if each agent is holding a different sample size, then the rate of concentration of random quantities associated to each agent (towards a population equivalent) will differ across the network. In order to utilise the concentration phenomena (for a linear speed-up in computational time) a bound on the difference between the minimum and maximum number of samples across the network required is then likely to be required. Intuitively, this is to ensure the error (from concentration) for the agent with the minimum number samples is below the optimal statistical error for the entire network (which depends on the largest number of samples). One possible relaxation is to require only a *subset* of agents to achieve a *near* optimal statistical rate with respect to all of

the samples within the network. For instance, we may allow the agents with fewer samples to perform worse versus, say, agents with larger sample sizes.

Stochastic Gradient Descent The work within Chapter 2 focused on Stochastic Gradient Descent, whereas Chapter 3 and 4 considered standard full-batch Gradient Descent. Extending the latter two chapters to stochastic gradients is natural as it is more computationally tractable when the sample size is large. Following the single-machine case [87], one approach is to introduce an additional error term that accounts for the sub-sampling error (from the stochastic gradients) at each agent within the network. In the single-machine case the analysis then hinges on the iterates minimising the Empirical Risk. Extending this aspect of the analysis to the distributed setting may pose a challenge as the iterates at each agent may not directly minimise the Empirical Risk with respect to their locally held samples.

Second Order methods and Sparsifying Communication Further investigation into the implicit regularisation effects of distributed algorithms as well as, following the centralised setting [65], development of other decentralised algorithms e.g. second order gradient methods, that exploit the statistical concentration of quantities held by agents can be performed. Moreover, since a speed-up can be achieved for a wider range of network topologies in a machine learning setting, the network may no longer be viewed as constraining communication between agents. This suggests communication savings through *sparsifying* the network topology by, say, having agents purposely not communicate to a sub-set of neighbours in order to save bandwidth.

The fifth chapter in this thesis, associated to the work "Tree-Based Multi-Task Sparse Recovery with Total Variation Penalty", explored a heterogeneous setting where the data held by agents in the network are drawn from similar, but not identical, sampling distributions. Specifically, a sparse recovery setting was considered where each agent wished to recover a sparse signal from potentially noisy linear measurements. We then considered a setting where the sparse signals associated to each agent were related in a manner reflecting the network topology. That is, if two agents were joined by an edge in the network, then the difference

between their underlying signals is sparse. This setting is then motivated by both hyper-spectral applications and distributed machine learning applications. Specifically, within hyper-spectral applications agents are associated pixels in a image, with the signal encoding the presence of a particular mineral or land type at any given pixel. The graph topology then encodes that neighbouring pixels are spatially correlated as, for instance, we expect the mineral composition to vary smoothly across the image. Meanwhile, for distributed machine learning, each agent is associated to a computer within a network wanting to fit a sparse linear model. As eluded to previously, due to the computers residing in different geographic locations, the data generating distribution may vary across the network. In this case the network topology can then play two roles: computational, encoding the communication channels between agents; and statistical, encoding the relationships between the data generating distributions associated to each agent.

Given this setting, we explored an approach to simultaneously recover all of the signals in a tree network by penalising the ℓ_1 norm of the signal root and the ℓ_1 total variation norm of the signals across the network. We showed theoretically and empirically, when the sparsity of the differences along edges are sufficiently small, that sample complexity savings can be achieved over other approaches that rely on the group lasso style penalties. This being due to our approach only requiring non-root agents to have their sample size scale with the sparsity *differences*, while guarantees for group lasso based methods require all of the agent's sample sizes to scale with the sparsity of their underlying signals. Moving forward, a case where the sparsity of the differences across edges in the network are not the same can be considered, in which case, a weighted total variation penalty can be investigated. Meanwhile, when the signals are sparse with respect to general network or graph, a number of different spanning tree topologies are appropriate, and thus, it is natural to investigate approaches that allow different trees to be compared.

References

[1] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *nature* 323.6088 (1986), pp. 533–536.

[2] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*. Vol. 1. 2. MIT press Cambridge, 2016.

[3] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4 (1989), pp. 541–551.

[4] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (2014).

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[8] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *arXiv preprint arXiv:2005.14165* (2020).

[9] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. "Understanding deep learning requires rethinking generalization". In: *arXiv preprint arXiv:1611.03530* (2016).

[10] Madhu S Advani, Andrew M Saxe, and Haim Sompolinsky. "High-dimensional dynamics of generalization error in neural networks". In: *Neural Networks* 132 (2020), pp. 428–446.

[11] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. "Surprises in high-dimensional ridgeless least squares interpolation". In: *arXiv preprint arXiv:1903.08560* (2019).

[12] Dominic Richards, Jaouad Mourtada, and Lorenzo Rosasco. "Asymptotics of Ridge (less) Regression under General Source Condition". In: *arXiv preprint arXiv:2006.06386* (2020).

[13] Denny Wu and Ji Xu. "On the Optimal Weighted ell_2 Regularization in Overparameterized Linear Regression". In: *arXiv preprint arXiv:2006.05800* (2020).

[14] Edgar Dobriban, Stefan Wager, et al. "High-dimensional asymptotics of prediction: Ridge regression and classification". In: *The Annals of Statistics* 46.1 (2018), pp. 247–279.

[15] Arthur Jacot, Franck Gabriel, and Clément Hongler. "Neural tangent kernel: Convergence and generalization in neural networks". In: *Advances in neural information processing systems*. 2018, pp. 8571–8580.

[16] Lenaic Chizat, Edouard Oyallon, and Francis Bach. "On lazy training in differentiable programming". In: *Advances in Neural Information Processing Systems*. 2019, pp. 2937–2947.

[17] Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. "Kernel and rich regimes in overparametrized models". In: *arXiv preprint arXiv:2002.09277* (2020).

[18] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Berlin, Heidelberg: Springer-Verlag, 1995.

[19] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*. Vol. 375. Springer Science & Business Media, 1996.

[20] Mathew Penrose. *Random geometric graphs*. 5. Oxford University Press, 2003.

[21] Piyush Gupta and Panganmala R Kumar. "The capacity of wireless networks". In: *IEEE Transactions on information theory* 46.2 (2000), pp. 388–404.

[22] A El Gamal, James Mammen, Balaji Prabhakar, and Devavrat Shah. "Throughput-delay trade-off in wireless networks". In: *IEEE INFOCOM 2004*. Vol. 1. IEEE. 2004.

[23] Alexandros DG Dimakis, Anand D Sarwate, and Martin J Wainwright. "Geographic gossip: Efficient averaging for sensor networks". In: *IEEE Transactions on Signal Processing* 56.3 (2008), pp. 1205–1216.

[24] David Kempe, Alin Dobra, and Johannes Gehrke. "Gossip-based computation of aggregate information". In: *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*. IEEE. 2003, pp. 482–491.

[25] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. "Gossip algorithms: Design, analysis and applications". In: *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*. Vol. 3. IEEE. 2005, pp. 1653–1664.

[26] John Nikolas Tsitsiklis. *Problems in decentralized decision making and computation*. Tech. rep. Massachusetts Inst Of Tech Cambridge Lab For Information and Decision Systems, 1984.

[27] Fan R.K. Chung and Fan Chung Graham. *Spectral graph theory*. 92. American Mathematical Soc., 1997.

[28] Devavrat Shah. *Gossip algorithms*. Now Publishers Inc, 2009.

[29] Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. "Optimal Algorithms for Smooth and Strongly Convex Distributed Optimization in Networks". In: *34th International Conference on Machine Learning*. PMLR, 2017, pp. 3027–3036.

[30] Kevin Scaman, Francis Bach, Sébastien Bubeck, Laurent Massoulié, and Yin Tat Lee. "Optimal algorithms for non-smooth distributed optimization in networks". In: *Advances in Neural Information Processing Systems*. 2018, pp. 2745–2754.

[31] Ohad Shamir and Nathan Srebro. "Distributed stochastic optimization and learning". In: *Communication, Control, and Computing (Allerton), 2014 52nd Annual Allerton Conference on*. IEEE. 2014, pp. 850–857.

[32] Angelia Nedic and Asuman Ozdaglar. "Distributed subgradient methods for multi-agent optimization". In: *IEEE Transactions on Automatic Control* 54.1 (2009), pp. 48–61.

[33] John C. Duchi, Alekh Agarwal, and Martin J. Wainwright. "Dual Averaging for Distributed Optimization: Convergence Analysis and Network Scaling". In: *IEEE Transactions on Automatic Control* 57.3 (2012), pp. 592–606.

[34] Rich Caruana. "Multitask learning". In: *Machine learning* 28.1 (1997), pp. 41–75.

[35] Dominic Richards and Patrick Rebeschini. "Graph-Dependent Implicit Regularisation for Distributed Stochastic Subgradient Descent". In: *Journal of Machine Learning Research* 21.2020 (2020), pp. 1–44.

[36] Dominic Richards and Patrick Rebeschini. "Optimal Statistical Rates for Decentralised Non-Parametric Regression with Linear Speed-up". In: *Advances in Neural Information Processing Systems*. 2019, pp. 1216–1227.

[37] Dominic Richards, Patrick Rebeschini, and Lorenzo Rosasco. "Decentralised learning with random features and distributed gradient descent". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 8105–8115.

[38] Angelia Nedic and Dimitri P Bertsekas. "Incremental subgradient methods for nondifferentiable optimization". In: *SIAM Journal on Optimization* 12.1 (2001), pp. 109–138.

[39] Ali Rahimi and Benjamin Recht. "Random features for large-scale kernel machines". In: *Advances in neural information processing systems*. 2008, pp. 1177–1184.

[40] Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. "A generalized representer theorem". In: *International conference on computational learning theory*. Springer. 2001, pp. 416–426.

[41] Marian-Daniel Iordache, José M Bioucas-Dias, and Antonio Plaza. "Sparse unmixing of hyperspectral data". In: *IEEE Transactions on Geoscience and Remote Sensing* 49.6 (2011), pp. 2014–2039.

[42] Marian-Daniel Iordache, José M Bioucas-Dias, and Antonio Plaza. "Total variation spatial regularization for sparse hyperspectral unmixing". In: *IEEE Transactions on Geoscience and Remote Sensing* 50.11 (2012), pp. 4484–4502.

[43] Xi Chen, Seyoung Kim, Qihang Lin, Jaime G Carbonell, and Eric P Xing. "Graph-structured multi-task regression and an efficient optimization method for general fused lasso". In: *arXiv preprint arXiv:1005.3579* (2010).

[44] Weiran Wang, Jialei Wang, Mladen Kolar, and Nathan Srebro. "Distributed Stochastic Multi-Task Learning with Graph Regularization". In: *arXiv preprint arXiv:1802.03830* (Feb. 2018).

[45] Simon Foucart and Holger Rauhut. "An invitation to compressive sensing". In: *A mathematical introduction to compressive sensing*. Springer, 2013, pp. 1–39.

[46] Oscar Hernan Madrid Padilla, James Sharpnack, James G Scott, and Ryan J Tibshirani. "The DFS Fused Lasso: Linear-Time Denoising over General Graphs." In: *J. Mach. Learn. Res.* 18 (2017), pp. 176–1.

[47] David Kempe, Alin Dobra, and Johannes Gehrke. "Gossip-Based Computation of Aggregate Information". In: (2003), pp. 482–. URL: http://dl.acm.org/citation.cfm?id=946243.946317.

[48] David Aldous and Jim Fill. *Reversible Markov chains and random walks on graphs*. 2002.

[49] Stephen Boyd, Persi Diaconis, and Lin Xiao. "Fastest mixing Markov chain on a graph". In: *SIAM review* 46.4 (2004), pp. 667–689.

[50] Ming Cao, Daniel A Spielman, and Edmund M Yeh. "Accelerated gossip algorithms for distributed computation". In: *Proc. of the 44th Annual Allerton Conference on Communication, Control, and Computation*. Citeseer. 2006, pp. 952–959.

[51] Patrick Rebeschini and Sekhar C Tatikonda. "Accelerated consensus via min-sum splitting". In: *Advances in Neural Information Processing Systems*. 2017, pp. 1374–1384.

[52] Mario Arioli and Jennifer Scott. "Chebyshev acceleration of iterative refinement". In: *Numerical Algorithms* 66.3 (2014), pp. 591–608.

[53] Stefania Sardellitti, Massimiliano Giona, and Sergio Barbarossa. "Fast distributed average consensus algorithms based on advection-diffusion processes". In: *IEEE Transactions on Signal Processing* 58.2 (2009), pp. 826–842.

[54] Raphaël Berthier, Francis Bach, and Pierre Gaillard. "Accelerated Gossip in Networks of Given Dimension using Jacobi Polynomial Iterations". In: *SIAM Journal on Mathematics of Data Science* 2.1 (2020), pp. 24–47.

[55] Fang Chen, László Lovász, and Igor Pak. "Lifting Markov chains to speed up mixing". In: *Proceedings of the thirty-first annual ACM symposium on Theory of computing*. ACM. 1999, pp. 275–281.

[56] John Tsitsiklis, Dimitri Bertsekas, and Michael Athans. "Distributed asynchronous deterministic and stochastic gradient optimization algorithms". In: *IEEE transactions on automatic control* 31.9 (1986), pp. 803–812.

[57] Bjorn Johansson, Maben Rabi, and Mikael Johansson. "A simple peer-to-peer algorithm for distributed optimization in sensor networks". In: *Decision and Control, 2007 46th IEEE Conference on*. IEEE. 2007, pp. 4705–4710.

[58] Angelia Nedić, Alex Olshevsky, Asuman Ozdaglar, and John N. Tsitsiklis. "On distributed averaging algorithms and quantization effects". In: *IEEE Transactions on Automatic Control* 54.11 (2009), pp. 2506–2517.

[59] Björn Johansson, Maben Rabi, and Mikael Johansson. "A randomized incremental subgradient method for distributed optimization in networked systems". In: *SIAM Journal on Optimization* 20.3 (2009), pp. 1157–1170.

[60] Ilan Lobel and Asuman Ozdaglar. "Distributed subgradient methods for convex optimization over random networks". In: *IEEE Transactions on Automatic Control* 56.6 (2011), pp. 1291–1306.

[61] Ion Matei and John S Baras. "Performance evaluation of the consensus-based distributed subgradient method under random communication topologies". In: *IEEE Journal of Selected Topics in Signal Processing* 5.4 (2011), pp. 754–771.

[62] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers". In: *Found. Trends Mach. Learn.* 3.1 (Jan. 2011), pp. 1–122.

[63] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. "Extra: An exact first-order algorithm for decentralized consensus optimization". In: *SIAM Journal on Optimization* 25.2 (2015), pp. 944–966.

[64] Aryan Mokhtari and Alejandro Ribeiro. "DSA: Decentralized double stochastic averaging gradient algorithm". In: *Journal of Machine Learning Research* 17.61 (2016), pp. 1–35.

[65] Ohad Shamir, Nathan Srebro, and Tong Zhang. "Communication-efficient distributed optimization using an approximate newton-type method". In: *International conference on machine learning*. 2014, pp. 1000–1008.

[66] Yuchen Zhang and Xiao Lin. "DiSCO: Distributed optimization for self-concordant empirical loss". In: *International conference on machine learning*. 2015, pp. 362–370.

[67] Yuchen Zhang, John Duchi, and Martin Wainwright. "Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates". In: *The Journal of Machine Learning Research* 16.1 (2015), pp. 3299–3340.

[68] Shao-Bo Lin, Xin Guo, and Ding-Xuan Zhou. "Distributed learning with regularized least squares". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 3202–3232.

[69] Zheng-Chu Guo, Shao-Bo Lin, and Ding-Xuan Zhou. "Learning theory of distributed spectral algorithms". In: *Inverse Problems* 33.7 (2017), p. 074009.

[70] Nicole Mücke and Gilles Blanchard. "Parallelizing spectrally regularized kernel algorithms". In: *The Journal of Machine Learning Research* 19.1 (2018), pp. 1069–1097.

[71] Junhong Lin and Volkan Cevher. "Optimal convergence for distributed learning with stochastic gradient methods and spectral algorithms". In: *Journal of Machine Learning Research* 21.147 (2020), pp. 1–63.

[72] Alekh Agarwal and John C. Duchi. "Distributed delayed stochastic optimization". In: *Advances in Neural Information Processing Systems*. 2011, pp. 873–881.

[73] Yuchen Zhang, Martin J. Wainwright, and John C. Duchi. "Communication-efficient algorithms for statistical optimization". In: *Advances in Neural Information Processing Systems*. 2012, pp. 1502–1510.

[74] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*. Vol. 87. Springer Science & Business Media, 2013.

[75] Olivier Bousquet and Léon Bottou. "The tradeoffs of large scale learning". In: *Advances in Neural Information Processing Systems*. 2008, pp. 161–168.

[76] Luc Devroye and Terry Wagner. "Distribution-free performance bounds for potential function rules". In: *IEEE Transactions on Information Theory* 25.5 (1979), pp. 601–604.

[77] Olivier Bousquet and André Elisseeff. "Stability and Generalization". In: *J. Mach. Learn. Res.* 2 (2002), pp. 499–526.

[78] Moritz Hardt, Benjamin Recht, and Yoram Singer. "Train Faster, Generalize Better: Stability of Stochastic Gradient Descent". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. 2016, pp. 1225–1234.

[79] Junhong Lin, Raffaello Camoriano, and Lorenzo Rosasco. "Generalization properties and implicit regularization for multiple passes SGM". In: *International Conference on Machine Learning*. 2016, pp. 2340–2348.

[80] Ilja Kuzborskij and Christoph Lampert. "Data-dependent stability of stochastic gradient descent". In: *arXiv preprint arXiv:1703.01678* (2017).

[81] Yuansi Chen, Chi Jin, and Bin Yu. "Stability and convergence trade-off of iterative optimization algorithms". In: *arXiv preprint arXiv:1804.01619* (2018).

[82] Liam Madden, Emiliano Dall'Anese, and Stephen Becker. "High probability convergence and uniform stability bounds for nonconvex stochastic gradient descent". In: *arXiv preprint arXiv:2006.05610* (2020).

[83] Andrea Caponnetto and Ernesto De Vito. "Optimal rates for the regularized least-squares algorithm". In: *Foundations of Computational Mathematics* 7.3 (2007), pp. 331–368.

[84] Yiming Ying and Massimiliano Pontil. "Online gradient descent learning algorithms". In: *Foundations of Computational Mathematics* 8.5 (2008), pp. 561–596.

[85] Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. "On early stopping in gradient descent learning". In: *Constructive Approximation* 26.2 (2007), pp. 289–315.

[86] Lorenzo Rosasco and Silvia Villa. "Learning with incremental iterative regularization". In: *Advances in Neural Information Processing Systems*. 2015, pp. 1630–1638.

[87] Junhong Lin and Lorenzo Rosasco. "Optimal rates for multi-pass stochastic gradient methods". In: *Journal of Machine Learning Research* 18.97 (2017), pp. 1–47.

[88] Nicolò Pagliana and Lorenzo Rosasco. "Implicit regularization of accelerated methods in hilbert spaces". In: *Advances in Neural Information Processing Systems* 32 (2019), pp. 14481–14491.

[89] Alexandre Défossez and Francis Bach. "Constant step size least-mean-square: Bias-variance trade-offs and optimal sampling distributions". In: *arXiv preprint arXiv:1412.0156* (2014).

[90] Aymeric Dieuleveut, Francis Bach, et al. "Nonparametric stochastic approximation with large step-sizes". In: *The Annals of Statistics* 44.4 (2016), pp. 1363–1399.

[91] Aymeric Dieuleveut, Nicolas Flammarion, and Francis Bach. "Harder, better, faster, stronger convergence rates for least-squares regression". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 3520–3570.

[92] Loucas Pillaud-Vivien, Alessandro Rudi, and Francis Bach. "Statistical Optimality of Stochastic Gradient Descent on Hard Learning Problems through Multiple Passes". In: *Advances in Neural Information Processing Systems 31*. 2018, pp. 8125–8135.

[93] Luigi Carratino, Alessandro Rudi, and Lorenzo Rosasco. "Learning with sgd and random features". In: *Advances in Neural Information Processing Systems*. 2018, pp. 10192–10203.

[94] Shun-ichi Amari, Jimmy Ba, Roger Grosse, Xuechen Li, Atsushi Nitanda, Taiji Suzuki, Denny Wu, and Ji Xu. "When Does Preconditioning Help or Hurt Generalization?" In: *arXiv preprint arXiv:2006.10732* (2020).

[95] Emmanuel J Candes and Terence Tao. "Decoding by linear programming". In: *IEEE transactions on information theory* 51.12 (2005), pp. 4203–4215.

[96] Berwin A Turlach, William N Venables, and Stephen J Wright. "Simultaneous variable selection". In: *Technometrics* 47.3 (2005), pp. 349–363.

[97] Joel A Tropp, Anna C Gilbert, and Martin J Strauss. "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit". In: *Signal processing* 86.3 (2006), pp. 572–588.

[98] Yuwon Kim, Jinseog Kim, and Yongdai Kim. "Blockwise sparse regression". In: *Statistica Sinica* 16.2 (2006), p. 375.

[99] Peng Zhao, Guilherme Rocha, Bin Yu, et al. "The composite absolute penalties family for grouped and hierarchical variable selection". In: *The Annals of Statistics* 37.6A (2009), pp. 3468–3497.

[100] Guillaume Obozinski, Ben Taskar, and Michael Jordan. *Joint covariate selection for grouped classification*. Tech. rep. Technical Report 743, Dept. of Statistics, University of California Berkeley, 2007.

[101] Sahand Negahban and Martin J Wainwright. "Joint support recovery under high-dimensional scaling: Benefits and perils of -regularization". In: *Proceedings of the 21st International Conference on Neural Information Processing Systems*. Curran Associates Inc. 2008, pp. 1161–1168.

[102] Guillaume Obozinski, Martin J Wainwright, Michael I Jordan, et al. "Support union recovery in high-dimensional multivariate regression". In: *The Annals of Statistics* 39.1 (2011), pp. 1–47.

[103] Ali Jalali, Sujay Sanghavi, Chao Ruan, and Pradeep K Ravikumar. "A dirty model for multi-task learning". In: *Advances in neural information processing systems*. 2010, pp. 964–972.