Testing and learning on distributional and set inputs

Ho Chung Leon Law

St Peter's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2019

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Ho Chung Leon Law

August 2019

Acknowledgements

I would like to thank everyone who has supported my DPhil study. In particular, I would like to give a special mention to my supervisor Professor Dino Sejdinovic for his patience and enthusiasm. I am always impressed by his knowledge, and how he can come up with so many innovative and exciting ideas. His guidance was of paramount importance especially in times when I am unsure about my direction and I am thankful to have him as my supervisor. There are also many colleagues that have helped me during my study, and I would like to give a huge thanks to Kaspar Märtens, Lucian Chan, Micheal Li, Jean-François Ton, Qinyi Zhang, Jin Xu, Anthony Caterini, Edwin Fong, Xiaoyu Lu, Yuan Zhou, Robert Hu, Alan Chau, Fan Wu and many others who have been with me. Without them, my life and my work would have been a lot less interesting, and of course, fewer excuses for tea breaks. I am also very thankful for the many enlightening discussions that we have had.

During my studies, I have collaborated with many people, all of whom have contributed greatly to my knowledge. I would like to thank Kenji Fukumizu, Christopher Yau, Dougal Sutherland, Seth Flaxman, Anant Raj, Mijung Park and many others for their patience and the many time spent together. A special thanks goes to my funding body EPSRC & MRC for providing me such a great opportunity. Outside of my academic work, there are also many that have influenced me and added colours to my life. I would like to thank my partner Joyce Yeung for her support, as well as my dear friends: Jacky Lee, Alex Tsui, Angel Wong, Timothy Wong, Crystal Leung, Taha Malik, Coco Yiu, Raphael Wong, Andrea Coladangelo and many others. Last but not the least, I am sincerely grateful to my family, who have constantly supported my education and goals. Without all of them, my experience at Oxford would not have been complete.

Abstract

As machine learning gains significant attention in many disciplines and research communities, the variety of data structures has increased, with examples including distributions and sets of observations. In this thesis, we consider sets and distributions as inputs for machine learning problems. In particular, we propose non-parametric tests, supervised learning, semi-supervised learning and meta-learning methodologies on these objects. In each case, with careful consideration of the input structure, we construct models that are applicable to various real life tasks.

We begin by considering the problem of *weakly supervised learning on aggregate outputs*, where the labels are only available at a much coarser resolution than the level of inputs, such that a set of inputs corresponds to each output. Constructing a tractable and scalable framework of aggregated observation models using Gaussian processes, we apply it to the important problem of fine-scale spatial modelling of malaria incidences. In particular, it is demonstrated that the prediction of unobserved pixel-level malaria intensities is possible using fine-scale environmental covariates.

Utilising the same data structure, but with the interpretation that the set of samples is drawn from a distribution, we consider the problem of modelling distributions in the context of hyperparameter selection for supervised learning tasks. Through transfer of information from previously solved tasks using learnt representations of the training datasets, we construct a Gaussian process framework that jointly models all the meta-information available. In application to a range of regression and classification tasks, we demonstrate that we achieve faster convergence compared to the state-of-the-art baselines.

Next we study the removal of noise on these distributional inputs. Specifically, building on advances in non-parametric deconvolution, we propose *phase features*, features of distributions that encode invariance to additive symmetric noise. Using such features and distances constructed with them, we construct novel nonparametric two-sample tests and methods for learning on distributional inputs. In both cases, as we encode invariance, we are able to target the underlying differences of interests in a variety of toy and real life applications. Finally, in a separate context, we consider the usage of noise to make the nonparametric two-sample test differentially private, i.e. to provide the protection of individual information in data analysis. Considering two settings of practical usage, we extend the current large-scale kernel two-sample test to be differentially private. Constructing an approximate finite-sample null distribution, we confirm that the protection of individual information is possible, while obtaining the correct Type I error with good power regimes.

The methodologies introduced in this thesis demonstrates how flexible statistical modelling of the underlying data structures can be brought to bear in tandem with performant machine learning algorithms. This opens up new research directions given the increasing variety of data structures, and therefore contributes to new applications in the machine learning community.

Contents

List of Figures

Chapter 1

Introduction

This thesis follows an integrated format and contains six chapters, with the first chapter as an introduction, and the last chapter as discussion. Each of the remaining chapters is based on a published paper and hence contains a literature review specific to the topics covered therein, whereas a more general introduction and overview is given in this chapter.

1.1 Motivation and contribution

Recent advances in computational power have allowed the development and usage of more complicated machinery in machine learning. Notably, this has resulted in major applications across areas of e-commerce, healthcare, computer vision, speech and language, and many more. Along with the advances in computational power, the cost of the data storage has decreased dramatically; this has led to an era of 'Big Data', with hundreds of terabytes of data being collected and stored. To make use of datasets of such sizes, deep learning [Goodfellow et al., 2016], a rich class of models that can model complicated non-linear transformations has been proposed. By stacking multiple levels of simple non-linear functions, and considering the dataset structure, we have seen successful applications in the fields such as computer vision, machine translation and reinforcement learning, to name a few.

The need to optimise these complicated models has led to the development of modern automatic differentiation libraries, like *TensorFlow* [Abadi et al., 2016], *PyTorch* [Paszke et al., 2017] and *MXNet* [Chen et al., 2015]. Through recursively applying simple rules of differentiation, these libraries can compute gradients and optimise any well-defined loss function,

allowing the training of complex models to be easily implemented. Furthermore, by making effective use of parallelisation with CPUs and GPUs, these libraries now allow training on data sizes that were deemed infeasible before.

As part of this 'Big Data' regime, we have also observed an increasing variety of data structures, with some of these examples including graphs [Narayanan et al., 2017], molecules [Dai et al., 2017], distributions [Szabó et al., 2016] and sets [Zaheer et al., 2017]. Focusing on the latter two as inputs, in this thesis, we will consider various frameworks to capture the underlying structure and extract those components that are useful for the problem at hand. For instance, in Chapter 2, we consider the problem of spatial mapping of diseases, where asymmetry in resolutions of inputs and outputs can lead to a whole set of covariates being associated to a single aggregated label. In this case, one important question is how to build a supervised model for non-aggregated labels using such data.

Additionally, in Chapter 3 we will consider the case of distributional inputs, i.e. where the sets of observations are assumed to be a random sample from a probability distribution. In this setting, we have the meta-learning scenario where each task contains a training dataset as input to a hyperparameter selection problem. By defining feature maps on these datasets, and treating input-output pairs as samples arising from a joint distribution, we can enable transfer of hyperparameter information to new tasks. Here, the question lies in how to learn feature maps that are invariant to variations that are not important for hyperparameter choice, and in particular how to define an appropriate framework to transfer relevant information from previous tasks.

A separate instance of distributional input modelling is the setting of distribution regression, which has seen applications in remote sensing [Wang et al., 2012], astronomy [Ntampaka et al., 2015, 2016] and prediction of the voting behaviour of demographic groups [Flaxman et al., 2015, 2016]. Here, the label is treated as an unknown function of the underlying distributional input, represented through its samples. As there are many sources of variability in real life, e.g. measurement noise, some questions that arise are, for example how to design a feature map of distributions that is robust to the impairment of the input distributions, and how we can use this feature map as part of the learning from distributions framework. Using the same formalism, we might also be interested in robust nonparametric two-sample testing,

with the goal of being able to target the underlying differences of interest. Alternatively, instead of the removal of noise, we may aim to utilise noise to protect individual information in a testing setting. In Chapter 4 and 5, we will discuss these formalisms and their connections.

In brief, the main contributions of the thesis can be can be summarised as follows:

- A framework for learning from aggregated outputs using Gaussian process. This method has important applications related to the spatial mapping of diseases.
- A joint Gaussian process model on hyperparameters and data representations, used for Bayesian optimisation (hyperparameter selection) in a meta-learning setting.
- A feature map of distribution that has symmetric noise invariance property. This feature map is applicable to robust two-sample testing and distribution regression.
- A kernel two-sample test that is differentially private, i.e. sensitive information about individuals is protected.

1.2 Background

As we will work with distributional and set inputs, we first define the necessary data structure for these inputs, correspondingly known as *bag data*. Let $\mathbf{x}_i^a \in \mathcal{X}$ be the i^{th} data-point in bag a, then a bag B_a is defined as a set with N_a data-points, with $B_a = \{\mathbf{x}_i^a\}_{i=1}^{N_a}$. If there exists a label for $\mathbf{x}_i^a$, then it is denoted by y_i^a. In the distributional setup, we will make an assumption that B_a are samples drawn from some distribution P_a; in the set case, we will simply treat it as a set of data-points. Additionally, depending on the scenario, for each B_a there may be a corresponding bag label y^a, which will be interpreted differently in these two settings.

1.2.1 Kernel methods

We begin by providing a general overview of the relevant concepts from the literature on kernel methods and reproducing kernel Hilbert spaces, as it is essential to constructing a feature map on distributions (see Muandet et al. [2017] for a more comprehensive overview). Suppose we have a positive definite function $k\colon \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ (with $\mathcal{X}$ as a generic non-empty set), then it can be shown that there exists a unique reproducing kernel Hilbert space

(RKHS) $\mathcal{H}_k$ of real-valued functions on $\mathcal{X}$ [Berlinet & Thomas-Agnan, 2011]. The k here is known as the kernel, with the function $k(\cdot, \mathbf{x})$ an element of $\mathcal{H}_k$ representing evaluation at $\mathbf{x}$ (reproducing property), i.e. $\langle f, k(\cdot, \mathbf{x})\rangle_{\mathcal{H}_k} = f(\mathbf{x})$, $\forall f \in \mathcal{H}_k$, $\forall\, \mathbf{x} \in \mathcal{X}$. Using the reproducing property, this suggests that for kernel $k(\mathbf{x}, \mathbf{y})$ there exists an implicit feature map $k(\cdot, \mathbf{x})$, such that $k(\mathbf{x}, \mathbf{y}) = \langle k(\cdot, \mathbf{x}), k(\cdot, \mathbf{y})\rangle_{\mathcal{H}_k}$. This establishes a correspondence between kernels and feature maps. Indeed, an alternative definition of a kernel is simply the inner product of explicit feature maps $\phi(\mathbf{x})$, with $k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y})\rangle_{\mathcal{H}_k}$ [Steinwart & Christmann, 2008].

As $\mathcal{X}$ is defined as any non-empty set, $k(\mathbf{x}, \mathbf{y})$ can provide a very flexible approach to measuring the similarity between two objects in any general space. In fact, through computation of $k(\mathbf{x}, \mathbf{y})$ directly, the computational costs associated with computing $\phi(\mathbf{x})$ (potentially infinite-dimensional) can be avoided. This is known as the 'kernel trick' and it is applicable to algorithms such as SVM [Schölkopf & Smola, 2002], PCA [Jolliffe, 2011] and any others that only require a similarity between data-points. However, it is noted that the computational cost of computing the kernel matrix is $\mathcal{O}(N^2)$, where N is the number of data-points, as computation of all pairwise similarities (i.e. kernel values) is necessary.

1.2.1.1 Translation invariant kernels and random Fourier features

Although many different kinds of kernels exist, in this thesis, we will mainly focus on the family of translation invariant kernels (i.e. $k(\mathbf{x}, \mathbf{y}) = \kappa(\mathbf{x} - \mathbf{y})$), with $\mathcal{X} = \mathbb{R}^d$. One such example is the automatic relevance determination (ARD) kernel, defined by:

$$k(\mathbf{x}, \mathbf{y}) = \gamma_{scale} \exp\left(-\frac{1}{2}\sum_{k=1}^{d}\frac{1}{\ell_k}(x_k - y_k)^2\right) \tag{1.1}$$

with scale parameter γ_{scale} and length scale parameters $\{\ell_k\}_{k=1}^d$. Here, its name arises because each individual length scale ℓ_k determines the 'relevance' for its corresponding dimension (where larger ℓ_k implies less variability across this dimension, hence less relevance). However in many cases we might take $\{\ell_k\}_{k=1}^d = \ell$ (i.e one length scale across all dimensions), which then recovers the widely used radial basis function (RBF) or Gaussian kernel.

Another translation invariant kernel commonly used in spatial statistics and Bayesian optimisation is the Matérn-$3/2$ kernel, defined by:

$$k(\mathbf{x}, \mathbf{y}) = \gamma_{scale} \left(1 + \sqrt{3} \sum_{k=1}^{d} \frac{(x_k - y_k)^2}{\ell_k}\right) \exp\left(-\sqrt{3} \sum_{k=1}^{d} \frac{(x_k - y_k)^2}{\ell_k}\right) \tag{1.2}$$

with scale parameter γ_{scale} and length scale parameters $\{\ell_k\}_{k=1}^{d}$.

One important theorem related to translation invariant kernels is the Bochner's theorem [Rudin, 1962]. It states that assuming a positive definite kernel $\kappa(\mathbf{x} - \mathbf{y})$ on $\mathbb{R}^d$ is scaled appropriately, its Fourier transform is a proper probability distribution $p(\boldsymbol{\omega})$, i.e.

$$k(\mathbf{x}, \mathbf{y}) = \kappa(\mathbf{x} - \mathbf{y}) = \int_{\mathbb{R}^d} p(\boldsymbol{\omega}) \exp\left(i\boldsymbol{\omega}^\top(\mathbf{x} - \mathbf{y})\right) d\boldsymbol{\omega} = \mathbb{E}_{\boldsymbol{\omega}}[\eta(\mathbf{x})\eta(\mathbf{y})^*] \tag{1.3}$$

where $\eta(\mathbf{x}) = \exp(i\boldsymbol{\omega}^\top\mathbf{x})$. As we most commonly work with real-valued $k(\mathbf{x}, \mathbf{y})$, we can obtain an unbiased estimation of $k(\mathbf{x}, \mathbf{y})$ by sampling from $p(\boldsymbol{\omega})$, i.e.

$$\hat{k}(\mathbf{x}, \mathbf{y}) = \frac{1}{J} \sum_{j=1}^{J} \left[\cos(\boldsymbol{\omega}_j^T\mathbf{x}) \cos(\boldsymbol{\omega}_j^T\mathbf{y}) + \sin(\boldsymbol{\omega}_j^T\mathbf{x}) \sin(\boldsymbol{\omega}_j^T\mathbf{y})\right] = \langle\phi(\mathbf{x}), \phi(\mathbf{y})\rangle_{\mathbb{R}^{2J}}, \tag{1.4}$$

with $\{\boldsymbol{\omega}_j\}_{j=1}^{J} \sim p(\boldsymbol{\omega})$ and $\phi(\mathbf{x}) = \sqrt{\frac{1}{J}}\left[\cos\left(\boldsymbol{\omega}_1^\top\mathbf{x}\right), \sin\left(\boldsymbol{\omega}_1^\top\mathbf{x}\right), \ldots, \cos\left(\boldsymbol{\omega}_J^\top\mathbf{x}\right), \sin\left(\boldsymbol{\omega}_J^\top\mathbf{x}\right)\right]$. This is known as random Fourier features (RFF)[1] and it has been first introduced in the influential paper by Rahimi & Recht [2007] who thus provided an approach for fast approximations to kernel methods. As we now work with a finite dimensional explicit feature map, the $\mathcal{O}(N^2)$ computation of the kernel matrix can often be avoided[2].

1.2.1.2 Kernel mean embedding

Throughout the thesis, we will often make the assumption that $B = \{\mathbf{x}_i\}_{i=1}^{N} \sim P$, i.e. each bag corresponds to random samples drawn from some underlying distribution. To explicitly model these distributions P, we will make use of kernel mean embeddings μ_P (cf. Smola et al. [2007] and Muandet et al. [2017] for a review), which serves as a high or infinite dimensional vector representation of P. To ensure $\mu_P \in \mathcal{H}_k$ is well-defined, it suffices that $\int \sqrt{k(\mathbf{x}, \mathbf{x})} dP(\mathbf{x}) < \infty$, which is satisfied for all P if k is bounded [Smola et al., 2007] (as

[1] Note that an alternative formulation of $\phi(\mathbf{x})$ is possible and is discussed in Sutherland [2016].

[2] Woodbury matrix identity can be used to bypass the explicit computation of the kernel matrix in many cases.

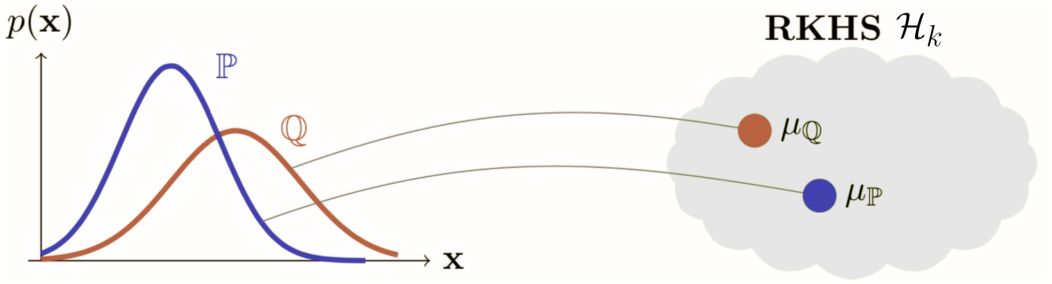

Figure 1.1: Embedding of marginal distributions via an expectation function, illustration is sourced from Muandet et al. [2017].

in the case of the ARD and Matérn kernel). Given a probability measure P on $\mathcal{X}$, we define the kernel mean embedding $\mu_P \in \mathcal{H}_k$ as follows:

$$\mu_P = \mathbb{E}_{X \sim P}[k(\cdot, X)] = \int_{\mathcal{X}} k(\cdot, \mathbf{x})\, dP(\mathbf{x}). \tag{1.5}$$

Analogous to the reproducing property of the RKHS, μ_P represents the expectation function on $\mathcal{H}_k$, i.e. $\int h(\mathbf{x}) dP(\mathbf{x}) = \langle h, \mu_P \rangle_{\mathcal{H}_k}$, with an illustration to be found in Figure 1.1. For so-called *characteristic* kernels [Sriperumbudur et al., 2010], every probability measure has a mapping between probability measures and their embeddings that is injective, and thus μ_P completely determines the corresponding probability measure. Examples include the ARD and Matérn kernel on $\mathbb{R}^d$.

In practice, as we often work with a set of samples $\{\mathbf{x}_i\}_{i=1}^N$ drawn from P instead of the direct measure, we will use an empirical estimator of μ_P, denoted $\widehat{\mu}_P \in \mathcal{H}_k$ and it is given by:

$$\widehat{\mu}_P = \mu_{\widehat{P}} = \int k(\cdot, \mathbf{x})\, d\widehat{P}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} k(\cdot, \mathbf{x}_i). \tag{1.6}$$

Using the reproducing property of $\mathcal{H}_k$, the following expression is readily derived:

$$\langle \widehat{\mu}_{P_a}, \widehat{\mu}_{P_b} \rangle_{\mathcal{H}_k} = \frac{1}{N_a N_b} \sum_{\ell=1}^{N_a} \sum_{r=1}^{N_b} k(\mathbf{x}_\ell^a, \mathbf{x}_r^b) \tag{1.7}$$

which gives us a notion of linear kernel on mean embeddings. Alternatively, we can define a Gaussian kernel on mean embeddings [Christmann & Steinwart, 2010]:

$$K(P_a, P_b) = \exp\left(-\frac{||\mu_{P_a} - \mu_{P_b}||^2_{\mathcal{H}_k}}{\ell} \right). \tag{1.8}$$

where ℓ is the corresponding lengthscale, and $||\mu_{P_a} - \mu_{P_b}||_{\mathcal{H}_k}$ is known as the maximum mean discrepancy (MMD) [Gretton et al., 2012a], a distance between distributions. Here, K is a kernel on probability measures that is characteristic [Christmann & Steinwart, 2010].

If there exists an explicit finite-dimensional feature map $\phi(\mathbf{x}) \in \mathbb{R}^J$ for kernel k, then we have a corresponding $k(\mathbf{x}, \mathbf{y})$ such that:

$$k(\mathbf{x}, \mathbf{y}) = \langle k(\cdot, \mathbf{x}), k(\cdot, \mathbf{y}) \rangle_{\mathcal{H}_k} = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_{\mathbb{R}^J} \tag{1.9}$$

in which case the empirical estimator $\widehat{\boldsymbol{\mu}}_P \in \mathbb{R}^J$ is simply given by:

$$\widehat{\boldsymbol{\mu}}_P = \frac{1}{N} \sum_{i=1}^{N} \phi(\mathbf{x}_i). \tag{1.10}$$

Through different representations of the feature map ϕ, we can define the features of the data distribution we would like to capture. For example $\phi(\mathbf{x}) = \mathbf{x}$ would be capturing the mean of the distribution P. Embedding joint and conditional distributions [Song et al., 2013; Fukumizu et al., 2013] is possible by considering operators on $\mathcal{H}_k$ and this will be discussed in Chapter 3.

1.2.1.3 Two-sample testing

One important application of kernel methods is in the area of nonparametric two-sample tests (cf. Gretton et al. [2012a] for a comprehensive treatment). Consider the following setting, where we have samples[3] $\{\mathbf{x}_i\}_{i=1}^N$ and $\{\mathbf{y}_i\}_{i=1}^N$, and $\{\mathbf{x}_i\}_{i=1}^N \sim P_X$ and $\{\mathbf{y}_i\}_{i=1}^N \sim P_Y$. We are interested in testing whether the null hypothesis $H_0 : P_X \stackrel{d}{=} P_Y$ is true, i.e. we would like to check whether the two distributions are equal, given only finite number of samples drawn from each distribution. To solve this problem, we consider a characteristic kernel k and consider the maximum mean discrepancy (MMD) [Gretton et al., 2012a] from (1.8) as the quantity of interest to be estimated:

$$\text{MMD}(P_X, P_Y) = \|\mu_{P_X} - \mu_{P_Y}\|_{\mathcal{H}_k}. \tag{1.11}$$

Note that this is a distance between distributions with kernel k, and the unbiased estimator of the squared MMD is given by:

$$\widehat{\text{MMD}}^2(P_X, P_Y) = \frac{1}{N(N-1)} \sum_{i \neq j} \left[k(\mathbf{x}_i, \mathbf{x}_j) + k(\mathbf{y}_i, \mathbf{y}_j) - k(\mathbf{x}_i, \mathbf{y}_j) - k(\mathbf{x}_j, \mathbf{y}_i) \right]. \tag{1.12}$$

[3] Alternative formulations are available when N differs between the two groups.

To estimate the asymptotic null distribution[4] [Gretton et al., 2012a, Theorem 12], there are two consistent approaches. One approach is to use the permutation test [Gretton et al., 2012a], which incurs a cost of $\mathcal{O}(N^2)$ per permutation, the alternative is to consider the spectral decomposition of the kernel matrix as in Gretton et al. [2009], which incurs a cost of $\mathcal{O}(N^3)$. To avoid these expensive computation costs, in this thesis, we will focus on an alternative formulation of (1.11) based on translation invariant and characteristic kernels, which has $\mathcal{O}(N)$ computational time and consistency against all alternatives.

Following Chwialkowski et al. [2015] and Jitkrittum et al. [2016], we now describe two linear time formulations (for translation invariant and characteristic kernels), with one using mean embeddings (ME) and the other employing an approach based on a smooth characteristic function (SCF). Both the ME and SCF tests consider finite-dimensional feature representations of the empirical measures $\widehat{P}_X$ and $\widehat{P}_Y$ corresponding to the samples $\{\mathbf{x}_i\}_{i=1}^N \sim P_X$ and $\{\mathbf{y}_i\}_{i=1}^N \sim P_Y$ respectively. The ME test considers feature representation given by $\widehat{\mu}_{P_X} = \frac{1}{N}\sum_{i=1}^N [k(\mathbf{x}_i, \mathbf{t}_1), \cdots, k(\mathbf{x}_i, \mathbf{t}_J)] \in \mathbb{R}^J$, for a given set of test locations $\{\mathbf{t}_j\}_{j=1}^J$, i.e. it evaluates the kernel mean embedding $\frac{1}{N}\sum_{i=1}^N k(\mathbf{x}_i, \cdot)$ of P_N at those locations. We take $\mathbf{w}_N = \widehat{\mu}_{P_X} - \widehat{\mu}_{P_Y}$ to be the difference of the feature vectors of the empirical measures of P_X and P_Y. Alternatively we can write

$$\mathbf{z}_i = \Big[k(\mathbf{x}_i, \mathbf{t}_1) - k(\mathbf{y}_i, \mathbf{t}_1), \cdots, k(\mathbf{x}_i, \mathbf{t}_J) - k(\mathbf{y}_i, \mathbf{t}_J)\Big] \tag{1.13}$$

then $\mathbf{w}_N = \frac{1}{N}\sum_{i=1}^N \mathbf{z}_i$. We also define the empirical covariance matrix to be

$$\mathbf{\Sigma}_N = \frac{1}{N-1}\sum_{i=1}^N (\mathbf{z}_i - \mathbf{w}_N)(\mathbf{z}_i - \mathbf{w}_N)^\top \tag{1.14}$$

with the final statistic given by

$$S_N = N\, \mathbf{w}_N^\top (\mathbf{\Sigma}_N + \gamma_N \mathbf{I})^{-1} \mathbf{w}_N. \tag{1.15}$$

Here a regularisation term $\gamma_N \mathbf{I}$ is added onto the empirical covariance matrix for numerical stability, with $\gamma_N \to 0$ as $N \to \infty$ [Jitkrittum et al., 2016]. The SCF setting uses the statistic

[4]Here consistency against all alternatives is guaranteed if a characteristic kernel is used.

of the same form, but considers features based on empirical characteristic functions [Rahimi & Recht, 2007]. Thus, it suffices to set $\mathbf{z}_i \in \mathbb{R}^J$ to

$$\mathbf{z}_i = \Big[g(\mathbf{x}_i)\cos(\mathbf{x}_i^\top \boldsymbol{\omega}_j) - g(\mathbf{y}_i)\cos(\mathbf{y}_i^\top \boldsymbol{\omega}_j), \\ g(\mathbf{x}_i)\sin(\mathbf{x}_i^\top \boldsymbol{\omega}_j) - g(\mathbf{y}_i)\sin(\mathbf{y}_i^\top \boldsymbol{\omega}_j) \Big]_{j=1}^{J/2}, \tag{1.16}$$

where $\{\boldsymbol{\omega}_j\}_{j=1}^{J/2}$ (J even) is a given set of frequencies, and g is a given function which has an effect of smoothing the characteristic function estimates (cf. Chwialkowski et al. [2015] for derivation and details). The test then proceeds in the same way as the ME version. For both cases, the distribution of the test statistic under the null hypothesis $H_0 : P_X \overset{d}{=} P_Y$ converges to a χ^2 distribution with J degrees of freedom. While Chwialkowski et al. [2015] uses random distribution features (i.e. test locations / frequencies are sampled randomly from a predefined distribution), Jitkrittum et al. [2016] selects test locations / frequencies which maximise the test power, yielding increased power and interpretable differences between the distributions under consideration.

1.2.1.4 Distribution regression

A separate application related to bag data in kernel methods is distribution regression [Sutherland, 2016; Szabó et al., 2016; Law et al., 2018b], i.e. supervised learning on distributions. In this setting, we have a dataset $\{\{\mathbf{x}_i^a\}_{i=1}^{N_a}, y^a\}_{a=1}^n$, where each $B_a = \{\mathbf{x}_i^a\}_{i=1}^{N_a}$ is assumed to be samples coming from some distribution P_a (implicitly drawn from some unknown meta-distribution over probability distributions), while y^a is the label per bag a (assumed to be a function of the underlying P_a, rather than any individual sample in the bag).

To obtain a feature representation for P_a, using the bag B_a, we can utilise the empirical kernel mean embedding, as described in (1.10):

$$\widehat{\mu}_{P_1} = \frac{1}{N_1}\sum_{i=1}^{N_1}\phi(\mathbf{x}_i^1), \;\; \ldots, \;\; \widehat{\mu}_{P_n} = \frac{1}{N_n}\sum_{i=1}^{N_n}\phi(\mathbf{x}_i^n) \tag{1.17}$$

where $\phi(\mathbf{x}) = k(\cdot, \mathbf{x}) \in \mathcal{H}_k$. Applying any positive definite kernel on $\mathcal{H}_k$ on these mean embeddings, e.g. linear kernel $K(B_a, B_b) = \langle \widehat{\mu}_{P_a}, \widehat{\mu}_{P_b} \rangle_{\mathcal{H}_k}$, we can perform classification [Muandet et al., 2012] or regression [Szabó et al., 2016] using (1.7), as we have now defined an appropriate similarity notion on distributions (through bag data).

Unfortunately, distribution regression as described using $\phi(\mathbf{x}) = k(\cdot, \mathbf{x}) \in \mathcal{H}_k$ is not scalable for even modestly-sized datasets, as computing each of the $\mathcal{O}(n^2)$ entries of the relevant kernel matrix is of $\mathcal{O}(N_a N_b)$. Hence, in practice we usually work with explicit finite dimensional feature map to compute empirical mean embeddings $\widehat{\boldsymbol{\mu}}_P \in \mathbb{R}^J$ as in (1.10). Given this representation of distribution in $\mathbb{R}^J$, we can use it as part of any standard supervised or unsupervised learning algorithms.

1.2.2 Gaussian processes and Variational inference

As we will be using various methodologies and applications from the Bayesian literature in Chapter 2 and 3, here we will provide a brief overview of the literature on Bayesian machine learning, and application of the framework of Gaussian processes and variational inference.

1.2.2.1 Variational inference

Bayesian inference corresponds to the computation of the posterior distribution $p(\mathbf{z}|\mathbf{x})$ of the unknown quantities $\mathbf{z}$ (which would include latent variables as well as model parameters) given the observations $\mathbf{x}$. In particular, using Bayes theorem, we can simply compute it by:

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})} \tag{1.18}$$

where here $p(\mathbf{z})$ is known as the prior and forms our initial belief about $\mathbf{z}$, while $p(\mathbf{x}|\mathbf{z})$ is known as the likelihood, and $p(\mathbf{x})$ as the model evidence. For complex Bayesian models, the posterior we are interested in is often intractable. This is why in variational inference [Jordan et al., 1999; Kingma & Welling, 2013; Blei et al., 2017], we will choose from a family of densities to approximate these posteriors, essentially converting the inference problem into an optimisation one. While it is possible to use MCMC methods [Gelman et al., 2013; Hastings, 1970; Geman & Geman, 1987; Neal et al., 2011] to sample from this posterior, in practice due to the complexity of the model or the size of the dataset, traditional MCMC methods might not be computationally feasible. Instead, considering a family of densities $\mathcal{F}$, variational inference solves the following problem:

$$q^*(\mathbf{z}) = \underset{q(\mathbf{z})\in\mathcal{F}}{\arg\min}\, \mathrm{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) \tag{1.19}$$

where KL represents the Kullback-Leibler divergence [Kullback & Leibler, 1951] on distributions:

$$\mathrm{KL}(q||p) = \int q(\mathbf{x}) \log \left(\frac{q(\mathbf{x})}{p(\mathbf{x})} \right) d\mathbf{x}. \tag{1.20}$$

Now this essentially implies that $q^*(\mathbf{z})$ is the member of $\mathcal{F}$ that is closest to the exact posterior in terms of the KL, and in variational inference, we take this to be the approximation to $p(\mathbf{z}|\mathbf{x})$. To optimise (1.19), we can consider its relation to the marginal likelihood[5] as defined by: $p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$. Using the definition of the KL divergence and expanding the conditionals, we can obtain:

$$\mathrm{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) = \log(p(\mathbf{x})) + \int q(\mathbf{z}) \log \left(\frac{q(\mathbf{z})}{p(\mathbf{z}, \mathbf{x})} \right) d\mathbf{z}. \tag{1.21}$$

Notice that as the direct computation of the KL of (1.19) is not possible, we instead consider (1.21) and notice that $\log(p(\mathbf{x}))$ is invariant with respect to $q(\mathbf{z})$. Hence, by maximising $\mathrm{ELBO}(q)$ defined by:

$$\begin{aligned} \mathrm{ELBO}(q) &= -\int q(\mathbf{z}) \log \left(\frac{q(\mathbf{z})}{p(\mathbf{z}, \mathbf{x})} \right) d\mathbf{z} \\ &= \int q(\mathbf{z}) \log p(\mathbf{x}|\mathbf{z}) d\mathbf{z} - \mathrm{KL}(q(\mathbf{z})||p(\mathbf{z})), \end{aligned} \tag{1.22}$$

we can minimise the KL divergence of (1.19). Here, this term is known as the evidence lower bound (ELBO), with its name arising from the fact that $\log(p(\mathbf{x})) \geq \mathrm{ELBO}(q)$ as the KL divergence is always ≥ 0. In summary, this suggests that by maximising the ELBO, a sum of the expect log likelihood of the data and the KL divergence between the prior $p(\mathbf{z})$ and $q(\mathbf{z})$, we can minimise the KL divergence between the approximation and the true posterior $p(\mathbf{z}|\mathbf{x})$.

When using variational inference, one important choice to be made is the family of densities $\mathcal{F}$, as this determines the accuracy of approximation, and the difficulty of optimisation. Here, we will review one such approach, known as the mean-field variational family, which considers the case where all the latent variables are independent, and governed by its individual density. Specifically, its form is given by:

$$q(\mathbf{z}) = \prod_{k=1}^{m} q_k(z_k) \tag{1.23}$$

[5]This is the normalising constant in the computation of the posterior, which is usually intractable and expensive.

where m is the dimension of the latent variable $\mathbf{z}$. Here q_k denotes the k^{th} variational distribution corresponding to the k^{th} dimension. In practice, q_k is fully determined by parameters $\boldsymbol{\theta}$ (e.g. m_k, s_k with $q_k = \mathcal{N}(m_k, s_k^2)$) which are then optimised in (1.22) using say Coordinate Ascent Variational Inference (CAVI) [Bishop, 2006]. For more information on the alternative formulations of $\mathcal{F}$ and scalable optimisation of (1.22), please refer to Hoffman et al. [2013], Blei et al. [2017] and Rezende & Mohamed [2015].

1.2.2.2 Gaussian process

Gaussian process (GP) is a Bayesian non-parametric approach that treats f, the function of interest, as a random variable in an infinite dimensional space of functions (cf. Rasmussen & Williams [2006] for a comprehensive treatment). It allows us to place a prior on functions, before updating it to a posterior on functions, after having observed its (noisy) evaluations at a set of points. In particular, a Gaussian process can be fully specified by its mean and covariance function, i.e.

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})] \qquad k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$$

then by definition for any set $\{\mathbf{x}_1, \dots \mathbf{x}_N\}$, we have that $[f(\mathbf{x}_1), \dots f(\mathbf{x}_N)]$ follows a multivariate normal distribution $\mathcal{N}(\mathbf{m}, \mathbf{K})$, with $\mathbf{m}_i = m(\mathbf{x}_i)$ and $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. For example, we can let $m(\mathbf{x})$ be a constant function, while $k(\mathbf{x}, \mathbf{x}')$ be the RBF kernel. Depending on the choice of the mean and covariance function, we can incorporate various prior knowledge on the distribution family of f (e.g. Chan et al. [2019]). Suppose now we have a GP prior on f, with observed data $\{\mathbf{x}_i, y_i\}_{i=1}^N$, and $\mathbf{f} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)]^\top$, then we can have the following regression model:

$$\begin{aligned} f &\sim \text{GP}(m(\cdot), k(\cdot, \cdot)) \\ \mathbf{f} &\sim \mathcal{N}(\mathbf{m}, \mathbf{K}_{\mathbf{xx}}) \\ \mathbf{y}|\mathbf{x} &\sim \mathcal{N}(\mathbf{f}, \sigma^2 \mathbf{I}) \end{aligned} \tag{1.24}$$

where $\mathbf{m}$ is a mean vector, $\mathbf{K}_{\mathbf{xx}}$ denotes the kernel matrix on inputs $\mathbf{x}$ and σ^2 is the variance of the noise. Using standard Gaussian conditioning, the posterior distribution is $\mathbf{f}|\mathbf{y} \sim$

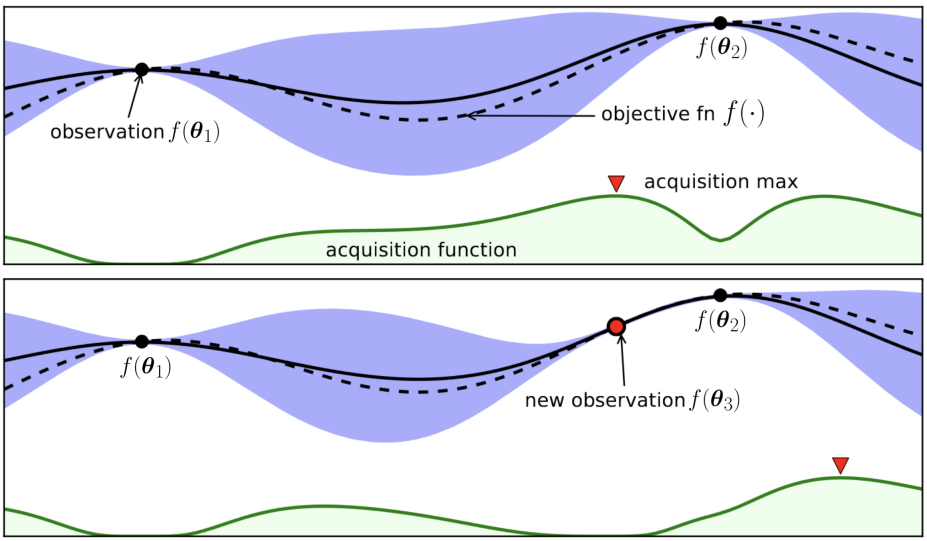

Figure 1.2: Illustration of Bayesian optimisation (without noise here), here the dotted line represents the underlying unknown f, while the non-dotted line refers to the Gaussian process model (with uncertainty represented by the shaded blue region). Figure is sourced from Brochu et al. [2010].

$\mathcal{N}(\mu_{\text{post}}, \Sigma_{\text{post}})$, where

$$\begin{aligned} \boldsymbol{\mu}_{\text{post}} &= \mathbf{m} + \mathbf{K}_{\mathbf{xx}}(\mathbf{K}_{\mathbf{xx}} + \sigma^2\mathbf{I})^{-1}(\mathbf{y} - \mathbf{m}) \\ \boldsymbol{\Sigma}_{\text{post}} &= \mathbf{K}_{\mathbf{xx}} - \mathbf{K}_{\mathbf{xx}}(\mathbf{K}_{\mathbf{xx}} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{\mathbf{xx}}. \end{aligned}$$

In addition, the posterior predictive distribution with test set $\mathbf{x}' = \{\mathbf{x}'_j\}_{j=1}^m$ is given by:

$$[f(\mathbf{x}'_1), \ldots f(\mathbf{x}'_m)] \sim \mathcal{N}(\mathbf{m}' + \mathbf{K}_{\mathbf{x}'\mathbf{x}}(\mathbf{K}_{\mathbf{xx}} + \sigma^2\mathbf{I})^{-1}(\mathbf{y} - \mathbf{m}), \mathbf{K}_{\mathbf{x}'\mathbf{x}'} - \mathbf{K}_{\mathbf{x}'\mathbf{x}}(\mathbf{K}_{\mathbf{xx}} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{\mathbf{xx}'}).$$

where $\mathbf{m}'$ denotes the mean function evaluation on $\mathbf{x}'$, $\mathbf{K}_{\mathbf{x}'\mathbf{x}}$ denotes the kernel matrix between $\mathbf{x}'$ and $\mathbf{x}$ and vice versa for $\mathbf{K}_{\mathbf{xx}'}$. As Gaussian process incurs the computational cost of $\mathcal{O}(N^3)$ due to the inversion, there are many existing approximations in the literature, some of which are reviewed in [Rasmussen & Williams, 2006, Chapter 8]. In Chapter 3, we will demonstrate the use of Bayesian linear regression [Bishop, 2006] as an approximation to GP.

1.2.2.3 Bayesian optimisation

In Bayesian optimisation (cf. Snoek et al. [2012] for a recent review), we have a function $f(\boldsymbol{\theta})$ that we want to optimise with respect to $\boldsymbol{\theta} \in \Theta$, i.e our goal is to find $\boldsymbol{\theta}^* = \text{argmax}_{\boldsymbol{\theta}\in\Theta} f(\boldsymbol{\theta})$. Now, as the function f is usually non-differentiable and non-convex with

respect to $\boldsymbol{\theta}$, we can only evaluate it in order to locate $\boldsymbol{\theta}^*$. One such example of f is the accuracy of a machine learning model with hyperparameter $\boldsymbol{\theta}$. While traditional methods, such as grid search and random search, can solve this problem, evaluation of f is often computationally expensive, hence we are interested in obtaining $\boldsymbol{\theta}^*$ in as few evaluations as possible.

Typically, as f is smoothly varying with respect to $\boldsymbol{\theta}$, by considering past evaluations of f, one can employ a sequential strategy to select $\boldsymbol{\theta}$. To do this, we will model f using a Gaussian process (GP) with a normal noise observation model, as described in (1.24). We now describe the typical steps involved in Bayesian optimisation (illustration can be found in Figure 1.2):

1. Independently sample at uniform s initial samples from Θ, and evaluate $f(\boldsymbol{\theta}_\ell)$ for $\ell = 1, \ldots, s$ to obtain pairs $\{(\boldsymbol{\theta}_\ell, z_\ell)\}_{\ell=1}^s$. Here z_ℓ represents a noisy observed evaluation of $f(\boldsymbol{\theta}_\ell)$.

2. Optimise any parameters of the GP through marginal likelihood maximisation and obtain the GP posterior with data $\{(\boldsymbol{\theta}_\ell, z_\ell)\}_{\ell=1}^s$.

3. Maximise the acquisition function $\alpha(\boldsymbol{\theta}; f)$ with respect to our fitted GP model on f to obtain $\boldsymbol{\theta}_{s+1}$.

4. Evaluate $f(\boldsymbol{\theta}_{s+1})$ to obtain z_{s+1}.

5. Repeat step 2 to step 4, adding newly evaluated observations to $\{(\boldsymbol{\theta}_\ell, z_\ell)\}_{\ell=1}^s$. Stop when computational cost is exhausted or the user-defined stopping rule is met.

6. Return $\boldsymbol{\theta}_{max} = \text{argmax}_{\boldsymbol{\theta}}\, z_\ell$ as an estimator of $\boldsymbol{\theta}^*$.

Here, the acquisition function $\alpha(\boldsymbol{\theta}; f)$ is a function of $\boldsymbol{\theta}$ that strikes the right trade-off between exploration and exploitation - intuitively we would like to explore areas of large uncertainty where large function values are still possible, but at the same time we would also like to exploit known information i.e. the current mean estimates of the function. One widely used acquisition function is the Expected Improvement (EI) [Močkus, 1975] given by:

$$\alpha(\boldsymbol{\theta}; f) = \mathbb{E}[u(\boldsymbol{\theta})|f], \text{with } u(\boldsymbol{\theta}) = \max(0, f(\boldsymbol{\theta}) - z_{max}) \tag{1.25}$$

where here z_{max} is the largest z_ℓ observed so far. We will discuss the use of this acquisition function further in Chapter 3.

1.3 Thesis outline

This thesis contains 4 papers, followed by conclusions and discussion on further work.

In Chapter 2, I will discuss the use of Gaussian process in learning from aggregated labels. This is a joint work with Dino Sejdinovic, Ewan Cameron, Tim Lucas, Seth Flaxman, Katherine Battle, and Kenji Fukumizu. The work was partly undertaken during my placement at the Institute of Statistical Mathematics, Tokyo under the supervision of Kenji Fukumizu. Together with Fukumizu and Sejdinovic, I contributed to the design of methods and formalisation of the mathematical framework and conducted all experiments myself (not including the collection and pre-processing of the malaria dataset). In addition, I derived all of the baseline formulations, as well as the precise model for the malaria incidence prediction problem. I also implemented the unpublished extension described in Section 2.8, and conducted additional experiments.

- **Ho Chung Leon Law**, Dino Sejdinovic, Ewan Cameron, Tim Lucas, Seth Flaxman, Katherine Battle, Kenji Fukumizu
  Variational learning on aggregate outputs with Gaussian processes
  https://papers.nips.cc/paper/7847-variational-learning-on-aggregate-outputs-with-gaussian-processes
  Advances in Neural Information Processing Systems (NeurIPS), 2018

Motivated by the application of spatial mapping of diseases, here we consider the problem of *weakly supervised learning on aggregates*. In this setting, while inputs can be collected cheaply at high resolution, this is not the case for the outputs which are collected at a much lower resolution. This asymmetry leads to a whole set of inputs being associated to a single aggregated output, which is assumed to be the sum of the corresponding label for each input observation in the set. Note that unlike in the distribution regression case, there is no assumption that samples in the set are drawn from a distribution; rather, our goal is to predict the label for each input data-point. Since we do not observe any labels at the same resolution as the inputs, we begin by defining a base model, before aggregating it. In this

work we propose the use of a Gaussian process to model the mean parameter of exponential families, before aggregation. Using variational inference, and additional approximations, we successfully implement this model on a real life malaria spatial mapping problem, with over 1 million observations.

In Chapter 3, I will investigate Bayesian optimisation in the setting where previous tasks have been solved. This is a joint work with Peilin Zhao, Lucian Chan, Junzhou Huang, and Dino Sejdinovic. Together with Sejdinovic, I contributed to the design of methods and formalisation of the mathematical framework and conducted all experiments myself (excluding the collection of the protein dataset). In particular, I derived the main formulation of distGP, and explored various approaches to embed the data distribution in the context of hyperparameter learning. Further, I proposed its scalable version distBLR, as well as formulated all baseline formulations, including manualBO, which is novel in this work. Lastly, I formulated the acquisition function and optimisation approach for the precise setting in the paper.

- **Ho Chung Leon Law**, Peilin Zhao, Lucian Chan, Junzhou Huang, Dino Sejdinovic
  Hyperparameter Learning via Distributional Transfer
  https://arxiv.org/abs/1810.06305
  Advances in Neural Information Processing Systems (NeurIPS), 2019.

In this work, the problem of interest is the selection of hyperparameters in machine learning models. Specifically, we consider the setting where a number of previous tasks have already been solved, and our goal is to transfer useful information to a new task. Unlike the previous chapter, we will assume that each task's training dataset consists of samples drawn from a joint distribution (though modelling as a set is also possible). Now by making use of the kernel mean embedding literature in Section 1.2.1.2 above, we construct an appropriate covariance function on the distribution of the training data, sample size of the training data and hyperparameters. By utilising this covariance function in a Gaussian process and Bayesian linear regression model, we construct a framework that can model all tasks simultaneously, enabling transfer of information from similar tasks. The resulting framework incorporates existing literature [Perrone et al., 2018; Swersky et al., 2013] in this setting. Lastly, its effectiveness is demonstrated across a range of regression and classification problems, including a real life protein-ligand binding problem in the area of drug design.

In Chapter 4, I will explore testing and learning regimes that are invariant to added symmetric noise in the distributional inputs. This is a joint work with Christopher Yau and Dino Sejdinovic. Together with Sejdinovic, I contributed to the design of methods and formalisation of the mathematical framework and conducted all experiments myself. In particular, I contributed to the construction of the SME and PhD two-sample test, as well as the approach for learning on distributions (including the learning of frequencies). Additionally, I also designed the toy and real life experiments, in order to demonstrate our methodology is indeed effective.

- **Ho Chung Leon Law**, Christopher Yau, Dino Sejdinovic
  Testing and learning on distributions with symmetric noise invariance
  https://papers.nips.cc/paper/6733-testing-and-learning-on-distributions-with-symmetric-noise-invariance
  Advances in Neural Information Processing Systems (NeurIPS), 2017.

In this work, we begin by discussing the construction of a feature map on distributions, which has invariant properties to any added symmetric positive definite (SPD) components, with examples including the zero mean Gaussian, Laplace and Cauchy distribution. The resulting feature map constructed is known as the *phase features*, and is connected to the random Fourier features through a simple normalisation procedure. Using such features, similar to the MMD, we construct a distance on distributions termed the *phase discrepancy* (PhD), which encodes invariances to these SPD components. The resulting distance has important applications in two-sample testing and distribution regression, as it allows for targeting of the underlying distribution of interests, rather than any SPD variations across the two samples. Empirically, we demonstrate these approaches are effective on a Higgs Boson search dataset, as well as problems related to aerosol and dark matter prediction.

In Chapter 5, I will discuss a kernel two-sample test that is differentially private [Dwork & Roth, 2014], i.e. the protection of individual information, while performing a nonparametric two-sample test. For this work, I am a joint first author with Anant Raj, and the work was supervised by Dino Sejdinovic and Mijung Park. Together with Raj and Park, I contributed to the framework of a differential private kernel two-sample test (excluding the proofs to be found in the appendix of Raj et al. [2019]). In addition, together with Sejdinovic, I

contributed to the formalisation for the correct control of the Type I error and conducted all experiments myself.

- Anant Raj*, **Ho Chung Leon Law***, Dino Sejdinovic, Mijung Park
  A Differentially Private Kernel Two-Sample Test
  https://arxiv.org/abs/1808.00380
  European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), 2019

Unlike the previous work where the main task is the *removal* of noise from the distributional input, here we *utilise* noise to protect sensitive individual information. Using the robust two-sample testing framework above, one natural idea is to simply add noise directly to the data, however this is not practical, as the level of noise needed for differential privacy can destroy the utility of the data (see Appendix 5.8.1 in Chapter 5). Hence, here we take an alternative approach and privatise only the quantities required. In particular, we propose two separate schemes for the privatisation of the kernel two-sample test, specifically targeting scenarios where there is a trusted third party and scenarios where there is no trustworthy entity between the two parties. Depending on the setting, the resulting methodology either perturbs the test statistic, or the mean and covariances of empirical differences of the data in a differentially private manner. While the asymptotic distribution under the null hypothesis remains the same in both cases, we show that in practice, this can lead to a grossly miscalibrated Type I error. To correct for this, we consider an approximation of the null distribution in the finite-sample regime. Empirically the proposed method provides improved Type I control and good power-privacy trade off.

* denote authors with equal contribution.

Chapter 2

Variational learning on aggregate outputs with Gaussian processes

This chapter is based on the following paper, apart from the material in Section 2.8 which is an extension and previously unpublished.

Ho Chung Leon Law, Dino Sejdinovic, Ewan Cameron, Tim Lucas, Seth Flaxman, Katherine Battle, and Kenji Fukumizu
Variational learning on aggregate outputs with Gaussian processes [Law et al., 2018a]
Advances in Neural Information Processing Systems (NeurIPS), 2018

While a typical supervised learning framework assumes that the inputs and the outputs are measured at the same levels of granularity, many applications, including global mapping of disease, only have access to outputs at a much coarser level than that of the inputs. Aggregation of outputs makes generalisation to new inputs much more difficult. We consider an approach to this problem based on variational learning with a model of output aggregation and Gaussian processes, where aggregation leads to intractability of the standard evidence lower bounds. We propose new bounds and tractable approximations, leading to improved prediction accuracy and scalability to large datasets, while explicitly taking uncertainty into account. We develop a framework which extends to several types of likelihoods, including the Poisson model for aggregated count data. We apply our framework to a challenging and important problem, the fine-scale spatial modelling of malaria incidence, with over 1 million observations.

2.1 Introduction

A typical supervised learning setup assumes existence of a set of pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ from which a functional relationship or a conditional probabilistic model of outputs given inputs can be learned. A prototypical use-case is the situation where obtaining outputs $y_\star$ for new, previously unseen inputs $\mathbf{x}_\star$ is costly, i.e. labelling is expensive and requires human intervention, but measurements of inputs are cheap and automated. Similarly, in many applications, due to a much greater cost in acquiring labels, they are only available at a much coarser resolution than the level at which the inputs are available and at which we wish to make predictions. This is the problem of *weakly supervised* learning on aggregate outputs [Kueck & de Freitas, 2005; Musicant et al., 2007], which has been studied in the literature in a variety of forms, with classification and regression notably being developed separately and without any unified treatment which can allow more flexible observation models. In this contribution, we consider a framework of observation models of aggregated outputs given bagged inputs, which reside in Exponential families. While we develop a more general treatment, the main focus in the chapter is on the Poisson likelihood for count data, which is motivated by the applications in spatial statistics.

In particular, we consider the important problem of fine-scale mapping of diseases. High resolution maps of infectious disease risk can offer a powerful tool for developing National Strategic Plans, allowing accurate stratification of intervention types to areas of greatest impact [Gething et al., 2016]. In low resource settings these maps must be constructed through probabilistic models linking the limited observational data to a suite of spatial covariates (often from remote-sensing images) describing social, economic, and environmental factors thought to influence exposure to the relevant infectious pathways. In this chapter, we apply our method to the incidence of clinical malaria cases.

Point incidence data of malaria is typically available at a high temporal frequency (weekly or monthly), but lacks spatial precision, being aggregated by administrative district or by health facility catchment. The challenge for risk modelling is to produce fine-scale predictions from these coarse incidence data, leveraging the remote-sensing covariates and appropriate regularity conditions to ensure a well-behaved problem.

Methodologically, the Poisson distribution is a popular choice for modelling count data. In the mapping setting, the intensity of the Poisson distribution is modelled as a function of spatial and other covariates. We use Gaussian processes (GP) as a flexible model for the intensity. GP is a widely used approach (reviewed in Section 1.2.2.2 in Chapter 1) in spatial modelling but also one of the pillars of Bayesian machine learning, enabling predictive models which explicitly quantify their uncertainty. Recently, we have seen many advances in variational GP posterior approximations, allowing them to couple with more complex observation likelihoods (e.g. binary or Poisson data [Nickisch & Rasmussen, 2008; Lloyd et al., 2015]) as well as a number of effective scalable GP approaches [Quiñonero Candela & Rasmussen, 2005; Titsias, 2009; Hensman et al., 2013, 2015], extending the applicability of GP to dataset sizes previously deemed prohibitive.

Contribution Our contributions can be summarised as follows. A general framework is developed for *aggregated observation models* using Exponential families and Gaussian processes. This is novel, as previous work on aggregation or bag models focuses on specific types of output models such as binary classification. Tractable and scalable variational inference methods are proposed for several instances of the aggregated observation models, making use of novel lower bounds on the model evidence. In experiments, it is demonstrated that the proposed methods can scale to dataset sizes of more than 1 million observations. We thoroughly investigate an application of the developed methodology to disease mapping from coarse measurements, where the observation model is Poisson, giving encouraging results. Uncertainty quantification, which is explicit in our models, is essential for this application. We also provide an extension of this work in Section 2.8, which is previously unpublished.

2.2 Related work

The framework of learning from aggregated data was believed to have been first introduced in Musicant et al. [2007], which considers the two regimes of classification and regression. However, while the task of classification of individuals from aggregated data (also known as *learning from label proportions*) has been explored widely in the literature [Quadrianto et al., 2009; Patrini et al., 2014; Kotzias et al., 2015; Melnikov & Hüllermeier, 2016; Yu et al., 2013, 2014; Kueck & de Freitas, 2005], there has been little literature on the analogous regression

regime in the machine learning community. Perhaps the closest literature available is Kotzias et al. [2015], who considers a general framework for learning from aggregate data, but also only considers the classification case for experiments. In this work, we will appropriately adjust the framework in Kotzias et al. [2015] and take this to be our baseline. A related problem arises in the spatial statistics community under the name of 'down-scaling', 'fine-scale modelling' or 'spatial disaggregation' [Keil et al., 2013; Howitt & Reynaud, 2003], in the analysis of disease mapping, agricultural data, and species distribution modelling, with a variety of proposed methodologies (cf. [Xavier et al., 2018] and references therein), including kriging [Goovaerts, 2010]. However, to the best of our knowledge, approaches making use of recent advances in scalable variational inference for GPs are not considered. It is noted that there is additional related work under the multi-task and multi-resolution setting [Hamelijnck et al., 2019], with the application of estimation of air pollution in London.

Another closely related topic is *multiple instance learning* (MIL), concerned with classification with max-aggregation over labels in a bag, i.e. a bag is positively labelled if at least one individual is positive, and it is otherwise negatively labelled. While the task in MIL is typically to predict labels of new unobserved *bags*, Haußmann et al. [2017] demonstrates that individual labels of a GP classifier can also be inferred in the MIL setting with variational inference. Our work parallels that approach, considering bag observation models in Exponential families and deriving new approximation bounds for some common generalised linear models. In deriving these bounds, we have taken an approach similar to Lloyd et al. [2015], who considers the problem of Gaussian process-modulated Poisson process estimation using variational inference. However, our problem is made more complicated by the aggregation of labels.

Other related research topics include distribution regression and set regression, as in Szabó et al. [2016], Law et al. [2017], Law et al. [2018b] and Zaheer et al. [2017]. In these regression problems, while the input data for learning is the same as the current setup, the goal is to learn a function at the bag level, rather than the individual level, the application of these methods in our setting, naively treating single individuals as 'distributions', may lead to sub-optimal performance. An overview of some other approaches for classification using bags of instances is given in Cheplygina et al. [2015].

2.3 Bag observation model: aggregation in mean parameters

Suppose we have a statistical model $p(y|\eta)$ for output $y \in \mathcal{Y}$, with parameter η given by a function of input $\mathbf{x} \in \mathcal{X}$, i.e., $\eta = \eta(\mathbf{x})$. Although one can formulate $p(y|\eta)$ in an arbitrary fashion, practitioners often only focus on interpretable simple models, hence we restrict our attention to $p(y|\eta)$ arising from Exponential families. To be more precise, the observation model is defined as:

$$p(y|\eta) = p(y|\theta) = \exp\left(\frac{y\theta - c(\theta)}{\tau}\right) h(y, \tau), \tag{2.1}$$

where response y is one-dimensional, θ is a natural parameter corresponding to the statistic y, τ is a dispersion parameter, and $h(y, \tau)$ is base measure[1]. Here η is the corresponding mean parameter, i.e.

$$\eta = \mathbb{E}_\theta[y] = \int y p(y|\theta) dy$$

and $\theta = F(\eta)$ be the link function (concave for all the examples here) mapping from mean to the natural parameters and $G(\theta)$ its inverse. In particular in this chapter, we will consider the following Exponential family models, with a particular focus on the Poisson and normal model:

- **Normal** (with fixed variance). $F = G = identity$ and there are no restrictions on the mean parameter space.
- **Poisson**. $F(\eta) = \log \eta$, $G(\theta) = e^\theta$ and η should take a positive value.
- **Exponential**. $p(y|\eta) = \exp(-y/\eta)/\eta$ and $\theta = -\eta$, $F(\eta) = -1/\eta$, $G(\theta) = -1/\theta$. Here η should take a positive value.

Given these observational models, assuming that now we have a fixed set of points $\mathbf{x}_i^a \in \mathcal{X}$ such that $B_a = \{\mathbf{x}_1^a, \ldots, \mathbf{x}_{N_a}^a\}$ is a *bag* of points with N_a *individuals*, and we wish to estimate the regression value $\eta(\mathbf{x}_i^a)$ for each individual. However, instead of the typical setup where we have a paired sample $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ of individuals and their outputs to use as a training set,

[1]For simplicity, we will assume that natural parameters corresponding to the other parts of the sufficient statistic are fixed and folded into the base measure.

we observe only *aggregate outputs* y^a for each of the bags. Hence, our training data is of the form:

$$(\{\mathbf{x}_i^1\}_{i=1}^{N_1}, y^1), \ldots (\{\mathbf{x}_i^n\}_{i=1}^{N_n}, y^n), \tag{2.2}$$

and the goal is to estimate parameters $\eta(\mathbf{x}_i^a)$ corresponding to individuals. To relate the aggregate y^a and the bag $B_a = \{\mathbf{x}_i^a\}_{i=1}^{N_a}$, we use the following *bag observation model*:

$$y^a | B_a \sim p(y|\eta^a), \qquad \eta^a = \sum_{i=1}^{N_a} w_i^a \eta(\mathbf{x}_i^a), \tag{2.3}$$

where w_i^a is an optional fixed non-negative weight used to adjust the scales (see Section 2.4 for an example). Note that the aggregation in the bag observation model can be on the mean parameters for individuals, not necessarily on the individual responses y_i^a. The reason for this approach is that if we consider a case where underlying individual responses y_i^a aggregate to y^a as a weighted sum, the form of the bag likelihood and individual likelihood would be different unless we restrict attention to distribution families which are closed under both scaling and convolution (such as Poisson and normal). However, when aggregation occurs in the mean parameter space, the form of the bag likelihood and individual likelihood is always the same. This implies the model for individual (unobserved) variables y_i^a ($i = 1, \ldots, N_a$) follow:

$$y_i^a \sim p(y|\eta_i^a), \qquad \eta_i^a = \Psi(f(\mathbf{x}_i^a)), \quad i = 1, \ldots, N_a, \; a = 1, \ldots, n. \tag{2.4}$$

In summary, this corresponds to the following measurement process:

- Each individual has a mean parameter η_i^a - if it were possible to sample a response for that particular individual, we would obtain a sample $y_i^a \sim p(\cdot|\eta_i^a)$.
- However, we cannot sample the individual and we can only observe a bag response. But in that case, only a single bag response is taken and depends on all individuals simultaneously. Each individual contributes in terms of an increase in a mean bag response, but this measurement process is different from the two-stage procedure by which we aggregate individual responses.

For tractable and scalable estimation, we will use variational methods, as the aggregated observation model leads to intractable posteriors when considering a Gaussian process model for f. While we devote a special focus to the Poisson and normal model in this chapter, the

formulation for the Exponential case can be found in Appendix 2.10.1, with an extension for the more general case to be found in Section 2.8. It is noted also that the bags can be of different sizes, and that after we obtain our individual model $\eta(\mathbf{x})$, we can use it for the prediction of in-bag individuals, as well as out-of-bag individuals.

2.4 Poisson bag model: Modelling aggregated counts

The Poisson distribution $p(y|\lambda) = \lambda^y e^{-\lambda}/(y!)$ is considered for count observations, and this chapter discusses the Poisson regression with intensity $\lambda(\mathbf{x}_i^a)$ multiplied by a 'population' p_i^a, which is a constant assumed to be known for each individual (or 'sub-bag') in the bag. The population for a bag a is given by $p^a = \sum_{i=1}^{N_a} p_i^a$. An observed bag count y^a is assumed to follow:

$$y^a|B_a \sim \mathrm{P}oisson(p^a\lambda^a), \quad \lambda^a := \sum_{i=1}^{N_a} \frac{p_i^a}{p^a}\lambda(\mathbf{x}_i^a).$$

Note that, by introducing unobserved counts $y_i^a \sim \mathrm{P}oisson(y_i^a|p_i^a\lambda(\mathbf{x}_i^a))$, the bag observation y^a has the same distribution as $\sum_{i=1}^{N_a} y_i^a$ since the Poisson distribution is closed under convolutions. If a bag and its individuals correspond to an area and its partition in geostatistical applications, as in the malaria example in Section 2.7.3, the population in the above bag model can be regarded as the population of an area or a sub-area. With this formulation, the goal is to estimate the basic intensity function $\lambda(\mathbf{x})$ from the aggregated observations (2.2). Assuming independence given $\{B_a\}_{a=1}^n$, the negative log-likelihood (NLL) ℓ_0 across bags is

$$\begin{aligned} -\log[\Pi_{a=1}^n p(y^a|B_a)] &\stackrel{c}{=} \sum_{a=1}^n p^a\lambda^a - y^a\log(p^a\lambda^a) \\ &\stackrel{c}{=} \sum_{a=1}^n \left[\sum_{i=1}^{N_a} p_i^a\lambda(\mathbf{x}_i^a) - y^a\log\left(\sum_{i=1}^{N_a} p_i^a\lambda(\mathbf{x}_i^a)\right)\right], \end{aligned} \tag{2.5}$$

where $\stackrel{c}{=}$ denotes an equality up to additive constant. During training, this term will pass information from the bag level observations $\{y^a\}_{a=1}^n$ to the individual basic intensity $\lambda(\mathbf{x}_i^a)$. It is noted that once we have trained an appropriate model for $\lambda(\mathbf{x}_i^a)$, we will be able to make individual level predictions, and also bag level predictions if desired. We will consider baselines with (2.5) using penalised likelihoods inspired by manifold regularisation in semi-supervised learning [Belkin et al., 2006] – presented in Section 2.6.

Proposing a model for λ based on GPs, we model f as a Gaussian process (GP), then we have:

$$y^a|B_a \sim \mathrm{P}oisson\left(\sum_{i=1}^{N_a} p_i^a\ \lambda_i^a\right), \qquad \lambda_i^a = \Psi(f(\mathbf{x}_i^a)), \qquad f \sim \mathrm{GP}(m(\cdot), k(\cdot,\cdot)) \quad (2.6)$$

where m and k are some appropriate mean[2] and covariance function. Since the intensity is always non-negative, in all models, we will need to use a transformation $\lambda(x) = \Psi(f(\mathbf{x}))$, where Ψ is a non-negative valued function. We will consider cases $\Psi(f) = f^2$ and $\Psi(f) = e^f$. A discussion of various choices of this link function in the context of Poisson intensities modulated by GP is given in Lloyd et al. [2015]. Modelling f with a GP allows us to propagate uncertainty on the predictions to λ_i^a, which is especially important in this weakly supervised problem setting, where we do not directly observe any individual output y_i^a. Since the total number of individuals in our target application of disease mapping is typically in the millions (see Section 2.7.3), we will approximate the posterior over $\lambda_i^a := \lambda(\mathbf{x}_i^a)$ using variational inference (reviewed in Section 1.2.2.1 of Chapter 1), with additional details to be found in Appendix 2.10.2.

For scalability of the GP method, as in the previous literature [Haußmann et al., 2017; Lloyd et al., 2015], we use a set of inducing points $\{\mathbf{u}_\ell\}_{\ell=1}^h$, which are given by the function evaluations of the Gaussian process f at landmark points $W = \{\mathbf{w}_\ell\}_{\ell=1}^h$; i.e. $\mathbf{u}_\ell = f(\mathbf{w}_\ell)$. The distribution $p(\mathbf{u}|W)$ is thus given by

$$\mathbf{u} \sim N(\boldsymbol{\mu}_W, \mathbf{K}_{WW}), \qquad \boldsymbol{\mu}_W = (m(\mathbf{w}_\ell))_\ell, \quad \mathbf{K}_{WW} = (k(\mathbf{w}_s, \mathbf{w}_t))_{s,t}. \quad (2.7)$$

The joint likelihood is given by:

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}|B, W, \Theta) = \prod_{a=1}^{n}\prod_{i=1}^{N_a} \mathrm{P}oisson(y^a|p^a\lambda^a)p(\mathbf{f}^a|\mathbf{u})p(\mathbf{u}|W), \text{ with } \mathbf{f}^a|\mathbf{u} \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_\mathbf{u}, \tilde{\mathbf{K}}), \quad (2.8)$$

$$\tilde{\boldsymbol{\mu}}_\mathbf{u}(\mathbf{z}) = \boldsymbol{\mu}_\mathbf{z} + \mathbf{k}_{\mathbf{z}W}\mathbf{K}_{WW}^{-1}(\mathbf{u} - \boldsymbol{\mu}_W), \quad \tilde{\mathbf{K}}(\mathbf{z}, \mathbf{z}') = k(\mathbf{z}, \mathbf{z}') - \mathbf{k}_{\mathbf{z}W}\mathbf{K}_{WW}^{-1}\mathbf{k}_{W\mathbf{z}'} \quad (2.9)$$

where $\mathbf{k}_{\mathbf{z}W} = [k(\mathbf{z}, \mathbf{w}_1), \ldots, k(\mathbf{z}, \mathbf{w}_\ell)]$, with $\boldsymbol{\mu}_W$, $\boldsymbol{\mu}_\mathbf{z}$ denoting their respective evaluations of the mean function $m(\cdot)$ and Θ being parameters of the mean and kernel functions of the GP. Here $B = \{B_a\}_{a=1}^n$ and implicitly $\tilde{\boldsymbol{\mu}}_\mathbf{u}, \tilde{\mathbf{K}}$ depends on B_a (according to the bag index on $\mathbf{f}^a$). It is also noted that $\mathbf{y}$ denotes $[y^1, \ldots, y^n]$, while $\mathbf{f} = [\mathbf{f}^1, \ldots, \mathbf{f}^n]$. Proceeding similarly

[2]For implementation, we consider a constant mean function.

to Lloyd et al. [2015], which discusses (non-bag) Poisson regression with GP, we obtain a lower bound of the marginal log-likelihood $\log p(\mathbf{y}|\Theta)$:

$$\begin{aligned}
\log p(\mathbf{y}|\Theta) &= \log \int\int p(\mathbf{y}, \mathbf{f}, \mathbf{u}|B, W, \Theta) d\mathbf{f} d\mathbf{u} \\
&\geq \int\int \log\Big\{p(\mathbf{y}|\mathbf{f}, \Theta)\frac{p(\mathbf{u}|W)}{q(\mathbf{u})}\Big\} p(\mathbf{f}|\mathbf{u}, \Theta) q(\mathbf{u}) d\mathbf{f} d\mathbf{u} \quad \text{(Jensen's inequality)} \\
&= \sum_a \int\int \Big\{ y^a \log\Big(\sum_{i=1}^{N_a} p_i^a \Psi(f(\mathbf{x}_i^a))\Big) - \Big(\sum_{i=1}^{N_a} p_i^a \Psi(f(\mathbf{x}_i^a))\Big)\Big\} p(\mathbf{f}^a|\mathbf{u}) q(\mathbf{u}) d\mathbf{f}^a d\mathbf{u} \\
&\quad - \sum_a \log(y^a!) - KL(q(\mathbf{u})||p(\mathbf{u}|W)) =: \mathcal{L}(q, \Theta), \qquad (2.10)
\end{aligned}$$

where $q(\mathbf{u})$ is a variational distribution to be optimised. The general solution to the maximisation over q of the evidence lower bound $\mathcal{L}(q, \Theta)$ above is given by the posterior of the inducing points $p(\mathbf{u}|\mathbf{y})$, which is intractable. We introduce a restriction to the class of $q(\mathbf{u})$ to approximate the posterior $p(\mathbf{u}|\mathbf{y})$. Suppose that the variational distribution $q(\mathbf{u})$ is Gaussian, $q(\mathbf{u}) = \mathcal{N}(\boldsymbol{\eta}_\mathbf{u}, \boldsymbol{\Sigma}_\mathbf{u})$. We then need to maximise the lower bound $\mathcal{L}(q, \Theta)$ over the variational parameters $\boldsymbol{\eta}_\mathbf{u}$ and $\boldsymbol{\Sigma}_\mathbf{u}$.

The resulting $q(\mathbf{u})$ gives an approximation to the posterior $p(\mathbf{u}|\mathbf{y})$ which also leads to a Gaussian approximation $q(\mathbf{f}^a) = \int p(\mathbf{f}^a|\mathbf{u}) q(\mathbf{u}) d\mathbf{u}$ to the posterior $p(\mathbf{f}^a|\mathbf{y})$, which we finally then transform through Ψ to obtain the desired approximate posterior on each $\lambda(\mathbf{x}_a^i)$ (which is either Log-normal or non-central χ^2 depending on the form of Ψ). The approximate posterior on λ will then allow us to make predictions for individuals while, crucially, taking into account the uncertainties in f (note that even the posterior predictive mean of λ will depend on the predictive variance in f due to the non-linearity Ψ). We also want to emphasis the use of inducing variables is essential for scalability in our model: we cannot directly obtain approximations to the posterior of $\lambda(\mathbf{x}_i^a)$ for all individuals, since this is often large in our problem setting (Section 2.7.3).

As the $p(\mathbf{u}|W)$ and $q(\mathbf{u})$ are both Gaussian, the last term (KL-divergence) of (2.10) can be computed explicitly with the exact form found in Appendix 2.10.2.3. To consider the first two terms, let $q^a(\mathbf{v}^a)$ be the marginal normal distribution of $\mathbf{v}^a = [f(\mathbf{x}_1^a), \ldots, f(\mathbf{x}_{N_a}^a)]$, where f follows the variational posterior $q^a = q(\mathbf{f}^a)$. The distribution of $\mathbf{v}^a$ is then $N(\mathbf{m}^a, \mathbf{S}^a)$, using

(2.9) :

$$\begin{aligned}\mathbf{m}^a &= \boldsymbol{\mu}_{B_a} + \mathbf{K}_{B_aW}\mathbf{K}_{WW}^{-1}(\boldsymbol{\eta}_{\mathbf{u}} - \boldsymbol{\mu}_W),\\ \mathbf{S}^a &= \mathbf{K}_{B_a,B_a} - \mathbf{K}_{B_aW}\left(\mathbf{K}_{WW}^{-1} - \mathbf{K}_{WW}^{-1}\boldsymbol{\Sigma}_{\mathbf{u}}\mathbf{K}_{WW}^{-1}\right)\mathbf{K}_{WB_a}.\end{aligned} \tag{2.11}$$

In the first term of (2.10), each summand is of the form

$$y^a \int \log\Big(\sum_{i=1}^{N_a} p_i^a \Psi\left(v_i^a\right)\Big) q^a(\mathbf{v}^a) d\mathbf{v}^a - \sum_{i=1}^{N_a} p_i^a \int \Psi\left(v_i^a\right) q^a(\mathbf{v}^a) d\mathbf{v}^a, \tag{2.12}$$

in which the second term is tractable for both of $\Psi(f) = f^2$ and $\Psi(f) = e^f$. The integral of the first term, however with q^a Gaussian is not tractable. To solve this, we take different approaches for $\Psi(f) = f^2$ and $\Psi(f) = e^f$; for the former, approximation by Taylor expansion is applied, while for the latter, further lower bound is taken. We also consider an alternative approach in Section 2.8.

First consider the case $\Psi(f) = f^2$, and rewrite the first term of (2.10) as:

$$y^a \mathbb{E}[\log \|\tilde{\mathbf{v}}^a\|^2] \quad \text{, where } \tilde{\mathbf{v}}^a \sim N(\tilde{\mathbf{m}}^a, \tilde{\mathbf{S}}^a),$$

with $\mathbf{P}^a = \text{diag}\left(p_1^a, \ldots, p_{N_a}^a\right), \tilde{\mathbf{m}}^a = (\mathbf{P}^a)^{1/2}\mathbf{m}^a$ and $\tilde{\mathbf{S}}^a = (\mathbf{P}^a)^{1/2}\mathbf{S}^a\mathbf{P}^{a1/2}$. By a Taylor series approximation for $\mathbb{E}[\log \|\tilde{\mathbf{v}}^a\|^2]$ (similar to Teh et al. [2007]) around $\mathbb{E}[\|\tilde{\mathbf{v}}^a\|^2] = \|\tilde{\mathbf{m}}^a\|^2 + tr(\tilde{\mathbf{S}}^a)$, we obtain

$$\begin{aligned}&\int \log\Big(\sum_{i=1}^{N_a} p_i^a (v_i^a)^2\Big) q^a(\mathbf{v}^a) d\mathbf{v}^a\\ &\quad\approx \log\left(\mathbf{m}^{a\top}\mathbf{P}^a\mathbf{m}^a + tr(\mathbf{S}^a\mathbf{P}^a)\right) - \frac{2\mathbf{m}^{a\top}\mathbf{P}^a\mathbf{S}^a\mathbf{P}^a\mathbf{m}^a + tr\Big((\mathbf{S}^a\mathbf{P}^a)^2\Big)}{\left(\mathbf{m}^{a\top}\mathbf{P}^a\mathbf{m}^a + tr(\mathbf{S}^a\mathbf{P}^a)\right)^2} =: \zeta^a.\end{aligned} \tag{2.13}$$

with details to be found in Appendix 2.10.2.2. An alternative approach which we use for the case $\Psi(v) = e^v$ is to take a further lower bound, which is applicable to a general class of Ψ. We use the following Lemma (proof found in Appendix 2.10.2.1):

Lemma 2.4.1. *Let $\mathbf{v} = [v_1, \ldots, v_N]^\top$ be a random vector with probability density $q(\mathbf{v})$ with marginal densities $q_i(\mathbf{v})$, and let $w_i \geq 0$, $i = 1, \ldots, N$. Then, for any non-negative valued function $\Psi(\mathbf{v})$,*

$$\int \log\Big(\sum_{i=1}^{N} w_i \Psi(v_i)\Big) q(\mathbf{v}) d\mathbf{v} \geq \log\Big(\sum_{i=1}^{N} w_i e^{\xi_i}\Big), \quad where \quad \xi_i := \int \log \Psi(v_i) q_i(v_i) dv_i.$$

Hence we obtain that

$$\int \log\Big(\sum_{i=1}^{N_a} p_i^a e^{v_i^a}\Big) q^a(\mathbf{v}^a) d\mathbf{v}^a \geq \log\Big(\sum_{i=1}^{N_a} p_i^a e^{m_i^a}\Big). \tag{2.14}$$

Using the above two approximation schemes, our objective (up to constant terms) can be formulated as:

1) $\Psi(v) = v^2$

$$\mathcal{L}^s(\Theta, \boldsymbol{\eta}_{\mathbf{u}}, \boldsymbol{\Sigma}_{\mathbf{u}}, W) := \sum_{a=1}^{n} y^a \zeta^a - \sum_{a=1}^{n} \sum_{i=1}^{N_a} \big\{(\mathbf{m}_i^a)^2 + \mathbf{S}_{ii}^a\big\} - KL(q(\mathbf{u})||p(\mathbf{u}|W)), \tag{2.15}$$

2) $\Psi(v) = e^v$

$$\mathcal{L}^e(\Theta, \boldsymbol{\eta}_{\mathbf{u}}, \boldsymbol{\Sigma}_{\mathbf{u}}, W) := \sum_{a=1}^{n} y^a \log\Big(\sum_{i=1}^{N_a} e^{\mathbf{m}_i^a}\Big) - \sum_{j=1}^{n} \sum_{i=1}^{N_a} e^{\mathbf{m}_i^a + \mathbf{S}_{ii}^a/2} - KL(q(\mathbf{u})||p(\mathbf{u}|W)). \tag{2.16}$$

Given these objectives, we can now optimise these lower bounds with respect to variational parameters $\{\boldsymbol{\eta}_{\mathbf{u}}, \boldsymbol{\Sigma}_{\mathbf{u}}\}$, parameters Θ of the mean and kernel functions, using stochastic gradient descent (SGD) on bags. Additionally, we might also learn W, locations for the landmark points. In this form, we can also see that the bound for $\Psi(v) = e^v$ has the added computational advantage of not requiring the full computation of the matrix $\mathbf{S}^a$, but only its diagonals, while for $\Psi(v) = v^2$ computation of ζ^a involves full $\mathbf{S}^a$, which may be problematic for extremely large bag sizes.

2.5 Normal bag model: Modelling aggregated observations

We will now consider the normal bag model for modelling aggregated observations. Similar to the Poisson distribution, the normal distribution is closed under both scaling and convolution, this implies that instead of considering the viewpoint of aggregation of mean parameters (described in Section 2.3), we can consider aggregated observations instead. The advantage of aggregation in this way is that we will be able to take account of variances of these observations in a more natural way. Consider a bag a of items $\{\mathbf{x}_i^a\}_{i=1}^{N_a}$, with item $\mathbf{x}_i^a$ assumed to have a weight w_i^a. At the individual level, we will model the (unobserved) responses y_i^a as $y_i^a|\mathbf{x}_i^a \sim \mathcal{N}\left(w_i^a \mu_i^a, (w_i^a)^2 \tau_i^a\right)$ where $\mu_i^a = f(\mathbf{x}_i^a)$ is the *mean parameter per unit weight* corresponding to the item $\mathbf{x}_i^a$ (a function of $\mathbf{x}_i^a$). Similarly, τ_i^a is the variance parameter per

unit weight. At the bag level, we consider the following model for the observed aggregated response $y^a = \sum_{i=1}^{N_a} y_i^a$, assuming conditional independence of individual responses given the covariates $B_a = \{\mathbf{x}_1^a, \ldots, \mathbf{x}_{N_a}^a\}$:

$$y^a | B_a \sim \mathcal{N}(w^a \mu^a, (w^a)^2 \tau^a) \text{ with } \mu^a = \sum_{i=1}^{N_a} \frac{w_i^a}{w^a} \mu_i^a,\ \tau^a = \frac{\sum_{i=1}^{N_a} (w_i^a)^2 \tau_i^a}{(w^a)^2} \tag{2.17}$$

where μ^a and τ^a are the mean and variance parameters per unit weight of the whole bag a and $w^a = \sum_{i=1}^{N_a} w_i^a$ is the *total weight* of bag a. Although we can take τ_i^a to also be a function of the covariates, here for simplicity, we take $\tau_i^a = \tau_a$ to be constant per bag (note the index of notation). We now compute the negative log-likelihood (NLL) across bags:

$$\begin{aligned} \ell_0 &= -\log\left[\Pi_{a=1}^n p(y^a | B_a)\right] \\ &= \frac{1}{2} \sum_{a=1}^{n} \left\{ \log\left(2\pi\tau_a \sum_{i=1}^{N_a} (w_i^a)^2\right) + \frac{\left(y^a - \sum_{i=1}^{N_a} w_i^a \mu_i^a\right)^2}{\sum_{i=1}^{N_a} (w_i^a)^2 \tau_a} \right\}. \end{aligned} \tag{2.18}$$

Here $\mu_i^a = f(\mathbf{x}_i^a)$ is the function we are interested in, and τ_a are the variance parameters to be learnt. Employing a Gaussian process model on f, using again an inducing point formulation, as in (2.6) and (2.7), we now consider the lower bound to the marginal likelihood as below (assuming $w_i^a = 1$ here to simplify notation, as the analogous expression with non-uniform weights is straightforward):

$$\begin{aligned} \log p(\mathbf{y}|\Theta) &= \log \int\int p(\mathbf{y}, \mathbf{f}, \mathbf{u} | B, W, \Theta) d\mathbf{f} d\mathbf{u} \\ &= \log \int\int \left(\prod_{a=1}^{n} \frac{1}{\sqrt{2\pi N_a \tau_a}} \exp\left(-\frac{(y^a - \sum_{i=1}^{N_a} f(\mathbf{x}_i^a))^2}{2N_a\tau_a} \right) \right) \frac{p(\mathbf{u}|W)}{q(\mathbf{u})} p(\mathbf{f}|\mathbf{u}) q(\mathbf{u}) d\mathbf{f} d\mathbf{u} \\ &\geq \int\int \log \left\{ \prod_{a=1}^{n} \frac{1}{\sqrt{2\pi N_a \tau_a}} \exp\left(-\frac{(y^a - \sum_{i=1}^{N_a} f(\mathbf{x}_i^a))^2}{2N_a\tau_a} \right) \frac{p(\mathbf{u}|W)}{q(\mathbf{u})} \right\} p(\mathbf{f}|\mathbf{u}) q(\mathbf{u}) d\mathbf{f} d\mathbf{u} \\ &= -\frac{1}{2} \sum_{a=1}^{n} \int\int \left\{ \frac{(y^a)^2 - 2y^a \sum_{i=1}^{N_a} f(\mathbf{x}_i^a) + \left(\sum_{i=1}^{N_a} f(\mathbf{x}_i^a)\right)^2}{N_a \tau_a} \right\} p(\mathbf{f}^a|\mathbf{u}) q(\mathbf{u}) d\mathbf{f}^a d\mathbf{u} \\ &\quad - \frac{1}{2} \sum_{a=1}^{n} \log(2\pi N_a \tau_a) - \int q(\mathbf{u}) \log \frac{q(\mathbf{u})}{p(\mathbf{u}|W)} d\mathbf{u}. \end{aligned} \tag{2.19}$$

Using a Gaussian distribution for $q(\mathbf{u}) = \mathcal{N}(\boldsymbol{\eta}_\mathbf{u}, \boldsymbol{\Sigma}_\mathbf{u})$, we have $q(\mathbf{f}^a) = \int p(\mathbf{f}^a|\mathbf{u}) q(\mathbf{u}) d\mathbf{u}$, which is a normal distribution. Now let $q(\mathbf{f}^a)$ be its marginal normal distribution of $\mathbf{f}^a = [f(\mathbf{x}_1^a), \ldots, f(\mathbf{x}_{N_a}^a)]$ with mean and covariance given by $\mathbf{m}^a$ and $\mathbf{S}^a$ as before in (2.11), then

all expectations with respect to $q(\mathbf{f}^a)$ are tractable and the ELBO is simply

$$\begin{aligned}\mathcal{L}(\Theta, \boldsymbol{\eta}_{\mathbf{u}}, \boldsymbol{\Sigma}_{\mathbf{u}}, W) = -\frac{1}{2}\sum_{a=1}^{n}\left\{\frac{(y^a)^2 - 2y^a\mathbf{1}^\top\mathbf{m}^a + \mathbf{1}^\top\left(\mathbf{S}^a + \mathbf{m}^a(\mathbf{m}^a)^\top\right)\mathbf{1}}{N_a\tau_a}\right\} \\ - \frac{1}{2}\sum_{a=1}^{n}\log(2\pi N_a\tau_a) - KL(q(\mathbf{u})||p(\mathbf{u}|W)).\end{aligned} \tag{2.20}$$

where $\mathbf{1}$ is $[1, \ldots, 1]^\top$. Given this objective, similar to the Poisson case, we can now optimise this with respect to the variational parameters $\{\boldsymbol{\eta}_{\mathbf{u}}, \boldsymbol{\Sigma}_{\mathbf{u}}\}$ and other any kernel hyperparameters using SGD on bags.

2.6 Alternative approaches

Here we will discuss various baselines and alternative approaches to the problem of weakly supervised learning on aggregate.

Constant We begin by considering the most simple baseline possible. Here, we simply take $\lambda_i^a = \lambda_c^a$, a constant rate across the bag, then:

$$\hat{\lambda}_c^a = \frac{y^a}{p^a}$$

hence the individual level predictive distribution is of the form $y_i^a \sim \text{Poisson}(\hat{\lambda}_c^a)$, and for unseen bag r, $\hat{\lambda}_c^{\text{bag}} = \frac{1}{\sum_{a=1}^n p^a}\sum_{a=1}^n y^a$, with predictive distribution $y^r \sim \text{Poisson}(p^r\hat{\lambda}_c^{\text{bag}})$. For the normal model, we can proceed in a similar fashion.

bag-pixel: Bag as individual Another baseline is to train a model from the weighted average of the covariates, given by $\mathbf{x}^a = \sum_{i=1}^{N_a}\frac{p_i^a}{p^a}\mathbf{x}_i^a$ in the Poisson case. The purpose of this baseline is to demonstrate that modelling at the individual level is important during training. Since we now have labels and covariates at the bag level, we can consider the following model for the Poisson case:

$$y^a|\mathbf{x}^a \sim \text{Poisson}(p^a\lambda(\mathbf{x}^a))$$

with $\lambda(\mathbf{x}^a) = \Psi(f(\mathbf{x}^a))$ for the Poisson model. For the normal model, we have:

$$y^a|\mathbf{x}^a \sim \mathcal{N}(w^a\mu(\mathbf{x}^a), (w^a)^2\tau)$$

where $\mu(\mathbf{x}^a) = f(\mathbf{x}^a)$ and τ is a parameter to be learnt (assumed constant across bags). Observing these models are in fact identical to the individual model except for a difference

in indexing, we can transfer the model to the individual level after learning f at the bag level. Essentially here we have created fake individual level instances by aggregation of individual covariates inside a bag.

Nyström: Bayesian MAP for regression on explicit feature maps Instead of the posterior based on the model (2.8), we can also consider an explicit feature map in order to directly construct a MAP estimator. While this method does not provide posterior uncertainty over λ_i^a, it does provide an interesting connection to the settings we have considered and also manifold-regularised neural networks, as discussed below. Let $\mathbf{K}_{\mathbf{zz}'}$ be the covariance function defined on covariates $\mathbf{z}$ and $\mathbf{z}'$, and consider its low rank approximation $\mathbf{K}_{\mathbf{zz}'} \approx \mathbf{k}_{\mathbf{z}W}\mathbf{K}_{WW}^{-1}\mathbf{k}_{W\mathbf{z}'}$ with landmark points $W = \{\mathbf{w}_\ell\}_{\ell=1}^h$, $\mathbf{k}_{\mathbf{z}W} = (k(\mathbf{z}, \mathbf{w}_h), \ldots, k(\mathbf{z}, \mathbf{w}_\ell))^T$ and similarly for $\mathbf{k}_{W\mathbf{z}'}$. By using landmark points W, we have avoided computation of the full kernel matrix, reducing computational complexity. Under this setup, we have that $\mathbf{K}_{\mathbf{zz}} \approx \mathbf{\Phi}_{\mathbf{z}}\mathbf{\Phi}_{\mathbf{z}}^\top$, with $\mathbf{\Phi}_{\mathbf{z}} = \mathbf{k}_{\mathbf{z}W}\mathbf{K}_{WW}^{-\frac{1}{2}}$ being the explicit (Nyström) feature map. Using this explicit feature map $\mathbf{\Phi}$, we have the following model (for the Poisson case):

$$f_i^a = \boldsymbol{\phi}_i^a \boldsymbol{\beta}, \qquad \boldsymbol{\beta} \sim \mathcal{N}(0, \gamma^2 I)$$
$$y^a | B_a \sim \mathrm{P}oisson\left(\sum_{i=1}^{N_a} p_i^a \lambda(\mathbf{x}_i^a)\right), \qquad \lambda(\mathbf{x}_i^a) = \Psi(f_i^a),$$

where γ is a prior parameter and $\boldsymbol{\phi}_i^a$ is the corresponding i^{th} row of $\mathbf{\Phi}_{B_a}$. For the normal case, we can proceed in a similar fashion. We now consider a MAP estimator of the model coefficients $\boldsymbol{\beta}$:

$$\hat{\boldsymbol{\beta}} = \mathrm{argmax}_{\boldsymbol{\beta}} \log[\Pi_{a=1}^n p(y^a | \boldsymbol{\beta}, B_a)] + \log p(\boldsymbol{\beta}). \tag{2.21}$$

This essentially recovers the same model as in (2.5) with the standard L_2 loss regularising the complexity of the function. This model can be thought of in several different ways, for example as a weight space view of the GP (cf. Rasmussen & Williams [2006] for an overview), or as a MAP of the Subset of Regressors (SoR) approximation [Smola & Bartlett, 2001] of the GP when $\sigma = 1$. Additionally, we may include manifold regularisation as part of the prior, as discussed below.

NN: Manifold-regularised neural networks The next approach we consider is a parametric model for f as in Kotzias et al. [2015], and search the best parameter to minimise the negative log-likelihood ℓ_0 (see (2.5) and (2.18)) across bags. Here we consider a neural

network with parameters $\boldsymbol{\theta}$ for the model f, and use back-propagation to learn $\boldsymbol{\theta}$ and hence individual level model f. However, since we only have aggregated observations at the bag level, but lots of individual covariate information, it is useful to incorporate this information by enforcing smoothness on the data manifold given by the unlabelled data. To do this, following Kotzias et al. [2015] and Patrini et al. [2014], we pursue a semi-supervised view of the problem and include an additional manifold regularisation term [Belkin et al., 2006] (re-scaled with N_{total}^2 during implementation):

$$\ell_1 = \sum_{w=1}^{N_{\text{total}}} \sum_{u=1}^{N_{\text{total}}} (f_u - f_w)^2 k_L(\mathbf{x}_u, \mathbf{x}_w) = \mathbf{f}^\top \mathbf{L}\, \mathbf{f} \tag{2.22}$$

where we have suppressed the bag index, N_{total} represents the total number of individuals, $k_{\mathbf{L}}(\cdot, \cdot)$ is some user-specified kernel[3], $\mathbf{f} = [f_1, \ldots, f_{N_{\text{total}}}]^\top$, $\mathbf{L}$ is the Laplacian defined as $\mathbf{L} = \text{diag}(\mathbf{K_L}\mathbf{1}^\top) - \mathbf{K_L}$, where $\mathbf{1}$ is just $[1, \ldots, 1]$ and $\mathbf{K_L}$ is the kernel matrix. Although this term involves calculation of a kernel matrix across individuals, in practice we consider stochastic gradient descent (SGD) and also random Fourier features [Rahimi & Recht, 2007] or Nyström approximation (see Appendix 2.10.3.1), with scale parameter λ_1 to control the strength of the regularisation. Similarly, one can also consider manifold regularisation at the bag level, if bag-level covariates/embeddings are available, for further details, see Appendix 2.10.3.2.

In fact, this same regularisation can be applied to the MAP estimation with the explicit feature maps and it is equivalent to having a prior $\boldsymbol{\beta} \sim \mathcal{N}(0, \sigma^2\mathbf{I} + (\lambda_1 \boldsymbol{\Phi}^\top \mathbf{L} \boldsymbol{\Phi})^{-1})$ that is data dependent and incorporates the structure of the manifold. For implementation, we consider a one hidden layer neural network with an output layer, for a fair comparison to the Nyström approach. For activation function, we consider the Rectified Linear Unit (ReLU).

2.7 Experiments

We will now demonstrate various approaches: Variational Bayes with GP (VBAgg), a MAP estimator of Bayesian Poisson regression with explicit feature maps (Nyström) and a neural network (NN) – the latter two employing manifold regularisation with RBF kernel (unless

[3]In practice, this can be derived from any notion of similarity between observations.

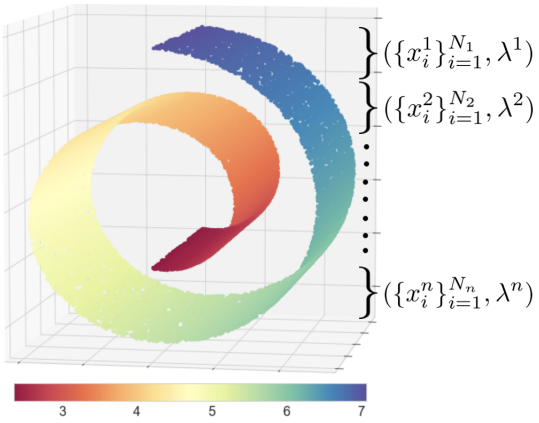

Figure 2.1: Random samples on the Swiss roll manifold.

stated otherwise). For additional baselines, we consider a constant within bag model (constant), i.e. $\hat{\lambda}_i^a = \frac{y^a}{p^a}$ and also consider creating 'individual' covariates by aggregation of the covariates within a bag (bag-pixel). We also denote $\Psi(v) = e^v$ and v^2 as Exp and Sq respectively. We implement our models[4] in *TensorFlow* [Abadi et al., 2016] and use ADAM [Kingma & Ba, 2015] to optimise their respective objectives, and we split the dataset into 4 parts, namely train, early-stop, validation and test set. Here the early-stop set is used for early stopping for the Nyström, NN and bag-pixel models, while the VBAgg approach ignores this partition as it optimises the lower bound to the marginal likelihood. The validation set is used for hyperparameter tuning of any regularisation scaling, as well as learning rate, layer size and multiple initialisations. Throughout, VBAgg and Nyström have access to the same set of landmarks for fair comparison. It is also important to highlight that we perform early stopping and tuning based on *bag* level performance on NLL only, as this is the only information available to us. For the VBAgg model, there are two approaches to tuning, one approach is to choose hyperparameters based on NLL on the validation bag sets, another approach is to select all hyperparameters based on the training objective $\mathcal{L}$, the lower bound to the marginal likelihood (denoted as VBAgg-Obj). In general, the results are relatively *insensitive* to this choice when $\Psi(v) = v^2$. To make predictions, we use the mean of our approximated posterior (provided by a log-normal and non-central χ^2 distribution for Exp and Sq). Additional information on the experimental setting can be found in Appendix 2.10.5.

[4]Code is available at https://github.com/hcllaw/VBAgg

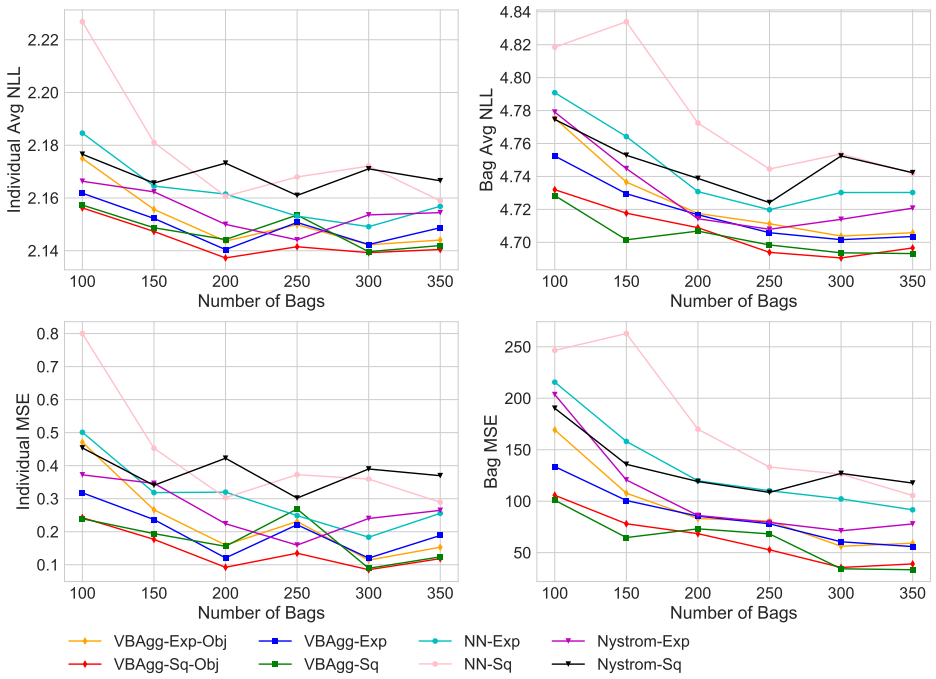

Figure 2.2: Varying number of bags over 5 repetitions. **Left Column:** Individual average NLL and MSE on train set. **Right Column:** Bag average NLL and MSE on test set (of size 500). Constant prediction NLL and MSE is 2.23 and 0.85 respectively. bag-pixel model prediction NLL is above 2.4 and MSE is above 3.0, hence not shown on graph.

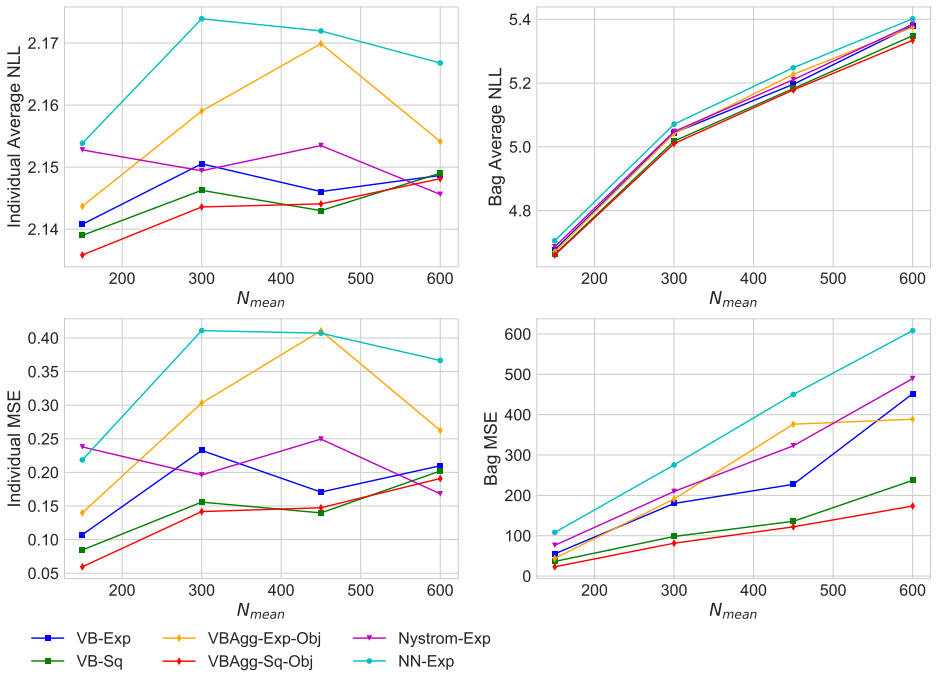

Figure 2.3: Varying number of individuals per bag N_{mean} over 5 repetitions. **Left Column:** Individual average NLL and MSE on train set. **Right Column:** Bag average NLL and MSE on test set (of size 500). Constant prediction NLL and MSE is 2.23 and 0.85 respectively.

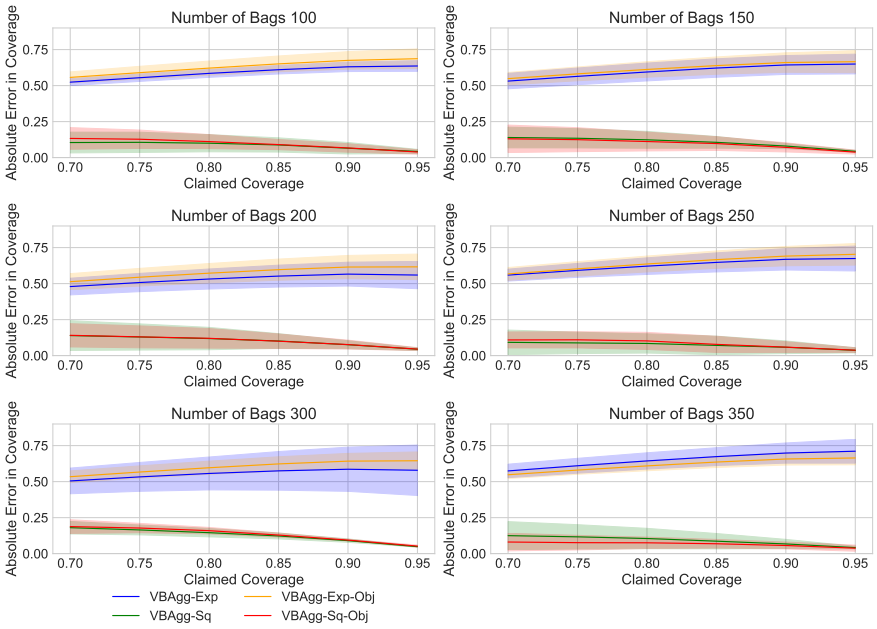

Figure 2.4: Absolute error in coverage from 70% to 95% for the increasing number of bags experiment for the Poisson model. Shaded regions highlight the standard deviation. Perfect coverage would provide a straight line at 0 error.

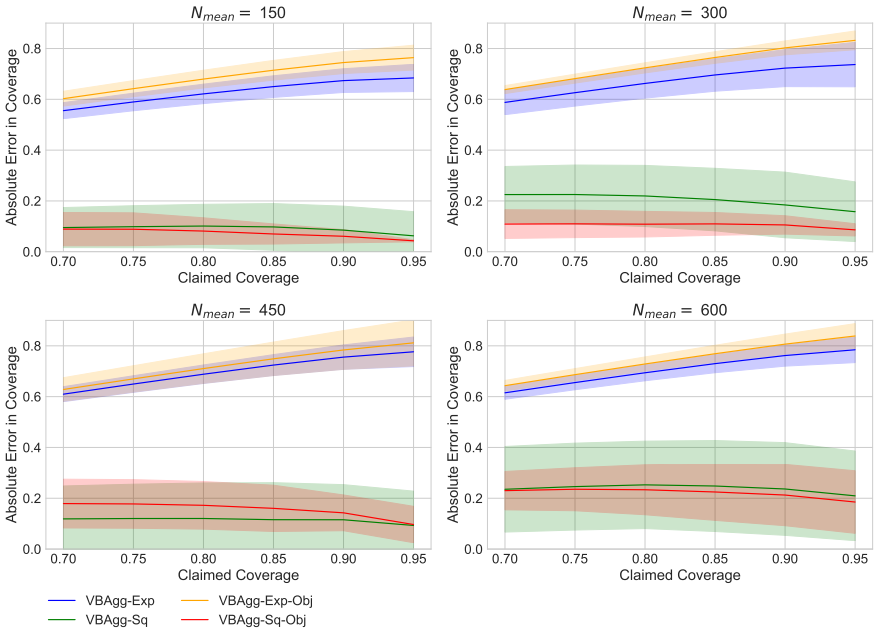

Figure 2.5: Absolute error in coverage from 70% to 95% for the increasing number of individuals per bag N_{mean} and N_{std} for the Poisson model. Shaded regions highlight the standard deviation. Perfect coverage would provide a straight line at 0 error.

2.7.1 Poisson model: Swiss roll

We first demonstrate our method on the swiss roll dataset[5], illustrated in Figure 2.1. To make this an aggregate learning problem, we first construct n bags with sizes drawn from a negative binomial distribution $N_a \sim NB(N_{mean}, N_{std})$, where N_{mean} and N_{std} represents the respective mean and standard deviation of N_a. We then randomly select $\sum_{a=1}^{n} N_a$ points from the swiss roll manifold to be the locations, giving us a set of coloured locations in $\mathbb{R}^3$. Ordering these random locations by their z-axis coordinate, we group them, filling up each bag in turn as we move along the z-axis. The aim of this is to simulate that in real life the partitioning of locations into bags is often not independent of covariates. Taking the colour of each location as the underlying rate λ_i^a at that location, we simulate $y_i^a \sim \text{Poisson}(\lambda_i^a)$, and take our observed outputs to be $y^a = \sum_{i=1}^{N_a} y_i^a \sim \text{Poisson}(\lambda^a)$, where $\lambda^a = \sum_{i=1}^{N_a} \lambda_i^a$. Our goal is then to predict the underlying individual rate parameter λ_i^a, given only bag-level observations y^a. To make this problem even more challenging, we embed the data manifold into $\mathbb{R}^{18}$ by rotating it with a random orthogonal matrix. For the choice of k for VBAgg and Nyström, we use the RBF kernel, with the bandwidth parameter learnt. For landmark locations, we use the K-means++ algorithm, so that landmark points lie evenly across the data manifold. In Appendix 2.10.5.2, we provide a similar analysis for the normal model, demonstrating that VBAgg outperforms other baselines.

Varying number of Bags: n To see the effect of increasing number of bags available for training, we fix $N_{mean} = 150$ and $N_{std} = 50$, and vary the number of bags n for the training set from 100 to 350 with the same number of bags for early stopping and validation. Each experiment is repeated for 5 runs, and results are shown in Figure 2.2 for individual and bag NLL, MSE on the train set. Again we emphasise that the individual labels are not used in training and that our goal is mainly focused on obtaining good individual NLL and MSE. We see that all versions of VBAgg outperform all other models, in terms of MSE and NLL, with statistical significance confirmed by a signed rank permutation test (see Appendix 2.10.5.1). We also observe that the bag-pixel model has poor performance, as a result of losing individual level covariate information in training by simply aggregating them.

[5]The swiss roll manifold function (for sampling) can be found in the Python *scikit-learn* package.

Varying number of individuals per bag: N_{mean} To study the effect of increasing bag sizes (with larger bag sizes, we expect 'disaggregation' to be more difficult), we fix the number of training bags to be 600 with early stopping and validation set to be 150 bags, while varying the number of individuals per bag through N_{mean} and N_{std} in the negative binomial distribution. To keep the relative scales between N_{mean} and N_{std} the same, we take $N_{std} = N_{mean}/2$. The results are shown in Figure 2.3, focusing on the best performing methods in the previous experiment. Here, we observe that VBAgg-Sq models perform better than the Nyström and NN models with statistical significance as reported in Appendix 2.10.5.1, with performance stable as N_{mean} increases. We also observe that the VBAgg-Sq models in general outperforms VBAgg-Exp models, and this is something we now discuss.

Discussion To gain more insight into the VBAgg model, we look at the calibration of our two different Bayesian models: VBAgg-Exp and VBAgg-Sq (and their Obj versions). We compute their respective posterior quantiles and observe the ratio of times the true λ_i^a lie in these quantiles. We present these in Figure 2.4 and 2.5. The calibration plots reveal an interesting nature about using the two different approximations for using e^v versus v^2 for $\Psi(v)$. While experiments showed that the two model perform similarly in terms of NLL (when tuning with NLL), the calibration of the models is very different. While the VBAgg-Sq is well calibrated in general, the VBAgg-Exp suffers from poor calibration. This is not surprising, as VBAgg-Exp uses an additional lower bound on model evidence, and this might also suggest why VBAgg-Exp-Obj that uses the lower bound for tuning purposes performs worse than VBAgg-Exp (as seen in Figure 2.3). Thus, the uncertainty estimates given by VBAgg-Exp should be treated with care.

2.7.2 Normal model: Elevators dataset

Having demonstrated that VBAgg outperforms other baselines for the normal model on the Swiss roll dataset in Appendix 2.10.5.2, we now consider a real life dataset. In particular, we consider the elevators dataset[6], which is a large scale regression dataset[7] containing 16599 instances, with each instance $\in \mathbb{R}^{17}$. This dataset is obtained from the task of controlling

[6]This dataset is publicly available at http://sci2s.ugr.es/keel/dataset.php?cod=94

[7]We have removed one column that is almost completely sparse.

Table 2.1: Results for the Normal model on the elevators dataset with 50 repetitions. Indiv represents individuals on train set here, while bag performance is measured on a test set. Numbers in brackets denotes p-values from a Wilcoxon signed-rank test for VBAgg versus the method. The null hypothesis is VBAgg performs equal or worse than NN or Nyström in terms of individual NLL or MSE on the train set. It is also noted MSE is computed on the observed y_i^a or y^a, rather than the unknown μ_i^a or μ^a, as they are unavailable.

	Indiv NLL	Bag NLL	Indiv MSE	Bag MSE
Constant	N/A	N/A	0.010	0.366
VBAgg	-1.69	0.003	0.0018	0.052
VBAgg-Obj	-1.71	-0.02	0.0018	0.052
Nyström	-1.57 (1.5e-13)	0.003	0.0024 (8.9e-16)	0.041
NN	-1.64 (0.0001258)	0.082	0.0021 (8.8e-10)	0.041

F16 aircraft, with the label y being a particular action taken on the elevators of the aircraft $\in \mathbb{R}$. For the model formulation we assume each label follows a normal distribution, i.e. $y_i \sim \mathcal{N}(\mu_i, \tau)$, where τ is a fixed quantity to be learnt. In practice, we can imagine the action taken may differ according to the operator.

In order to formulate this dataset in an aggregated data setting, we sample bag sizes from a negative binomial distribution as before, with $N_{mean} = 30$ and $N_{std} = 15$, and also take $w_i^a = 1$. To place observations into bags, similar to the swiss roll dataset, we consider a particular covariate, and place instances into bags based on the ordering of the covariate. We now have the bag-level model given by $y^a \sim \mathcal{N}(\mu^a, N_a\tau)$, with individual model $y_i^a \sim \mathcal{N}(\mu_i^a, \tau)$ and it is our goal to predict μ_i^a (and also infer τ), given only y^a and B_a. After the bagging process, we obtain approximately 225 bags for training, and 33 bags each for early stopping, validation and testing (for bag level performance). Further, in order to neglect variables that do not provide signal, we use an ARD (automatic relevance determination) kernel for the VBAgg and Nyström model, as below:

$$k_{ard}(\mathbf{x}, \mathbf{y}) = \gamma_{scale} \exp\left(-\frac{1}{2}\sum_{k=1}^{16}\frac{1}{\ell_k}(x_k - y_k)^2\right) \tag{2.23}$$

and learn kernel parameters γ_{scale} and $\{\ell_k\}_{k=1}^d$. We repeat this process and splitting of the dataset 50 times and report individual and Bag NLL, MSE results in Table 2.1. From the results, we observe that the VBAgg model performs better the Nyström and NN model, with statistical significance.

Table 2.2: Results for the Poisson model on the malaria dataset with 10 different re-splits of the data. Bag performance is measured on a test set, with MSE computed between $\log(y^a)$ and $\log(\sum_{i=1}^{N_a} p_i^a \hat{\lambda}_i^a)$ as the underlying true rates are not known. Brackets include standard deviation.

	Bag NLL	Bag MSE (Log)
Constant	173.1 (31.2)	4.08 (0.13)
Nyström-Exp	88.1 (25.1)	1.31 (0.15)
VBAgg-Sq-Obj	94.1 (34.0)	1.21 (0.05)
VBAgg-Exp-Obj	97.2 (39.6)	1.04 (0.11)
VBAgg-Sq	97.6 (39.0)	1.38 (0.18)
VBAgg-Exp	99.2 (39.8)	1.21 (0.19)
NN-Exp	164.4 (127.8)	1.82 (0.29)

2.7.3 Poisson model: Malaria incidence prediction

We now demonstrate the proposed methodology on an important real life malaria prediction problem for an endemic country from the Malaria Atlas Project database[8]. In this problem, we would like to predict the underlying malaria incidence rate in each 1km by 1km region (referred to as a pixel), while having only observed aggregated incidences of malaria y^a at much larger regional levels, which are treated as bags of pixels. These bags are non-overlapping administrative units, with N_a pixels per bag ranging from 13 to 6,667, with a total of 1,044,683 pixels. In total, data is available for 957 bags[9]. Along with these pixels, we also have population estimates p_i^a (per 1000 people) for pixel i in bag a, spatial coordinates given by $\mathbf{s}_i^a$, as well as covariates $\mathbf{x}_i^a \in \mathbb{R}^{18}$, collected by remote sensing. Some examples of covariates includes accessibility, topographic wetness index, distance to water, mean of land surface temperature and stable night lights. It is clear that rather than expecting malaria incidence rate to be constant throughout the entire bag (left of Figure 2.6), we expect pixel incidence rate to vary, depending on social, economic and environmental factors [Weiss et al., 2015], such as those in the middle and right of Figure 2.6. Our goal is therefore to build models that can predict malaria incidence rates at a *pixel* level.

We assume a Poisson model on each individual pixel, i.e. $y^a \sim \text{Poisson}(\sum_i p_i^a \lambda_i^a)$, where λ_i^a is the underlying pixel incidence rate of malaria per 1000 people that we are interested

[8]Due to confidentiality reasons, we do not report the country or plot the full map of our results.

[9]We consider 576 bags for train, 95 bags each for validation and early-stop, with 191 bags for testing, with different splits across different trials, selecting them to ensure distributions of labels are similar across sets.

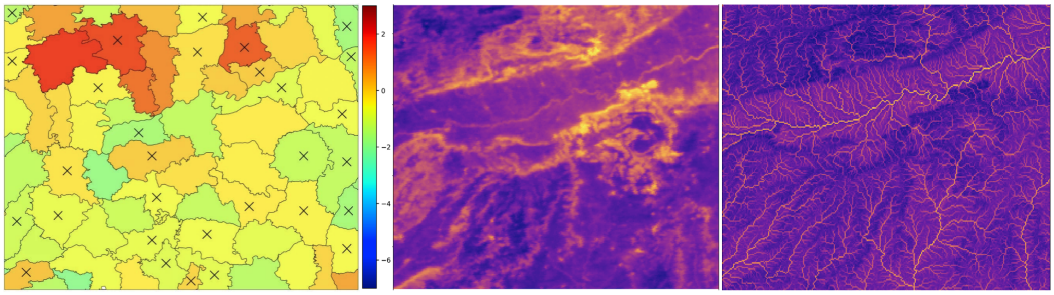

Figure 2.6: **Left**: Log of malaria incidence rate λ_i^a per 1000 people with constant model. Crosses denotes non-training bags. **Middle**: Topographic wetness index, lighter colours are wetter. **Right**: Land surface temperature at night, lighter colours are hotter.

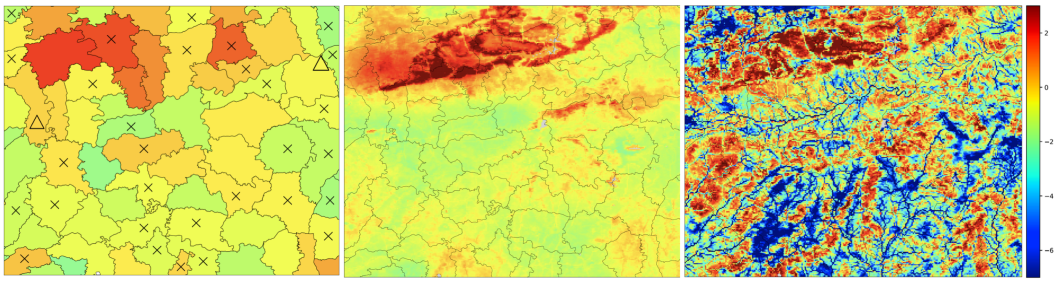

Figure 2.7: Log of malaria incidence rate λ_i^a per 1000 **Left**: Constant **Middle**: Nyström-Exp **Right**: NN-Exp

in predicting. We consider the VBAgg, Nyström and NN as prediction models and use the following kernel:

$$k((\mathbf{x}, \mathbf{s}_x), (\mathbf{y}, \mathbf{s}_y)) = \gamma_1 \exp\left(-\frac{1}{2}\sum_{k=1}^{18}\frac{1}{\ell_k}(x_k - y_k)^2\right) + \gamma_2\left(1 + \frac{\sqrt{3}||\mathbf{s}_x - \mathbf{s}_y||_2}{\rho}\right)\exp\left(-\frac{\sqrt{3}||\mathbf{s}_x - \mathbf{s}_y||_2}{\rho}\right) \quad (2.24)$$

which is simply a sum of an ARD kernel on covariates (with parameters $\gamma_1, \{\ell_k\}_{k=1}^{18}$) and a Matérn kernel on spatial locations (with parameters γ_2, ρ). The kernel parameters can be learnt here, and for the NN model we use the same kernel for manifold regularisation. This kernel choice incorporates spatial information, while allowing feature selection amongst other covariates. For choice of landmarks, we ensure landmarks are placed evenly throughout space by using one landmark point per training bag (selected by k-means++). This is so that the uncertainty estimates we obtain are not too sensitive to the choice of landmarks.

In this problem, no individual-level labels are available, so we report Bag NLL and MSE (on observed incidences) on the test bags in Table 2.2 over 10 different re-splits of the data.

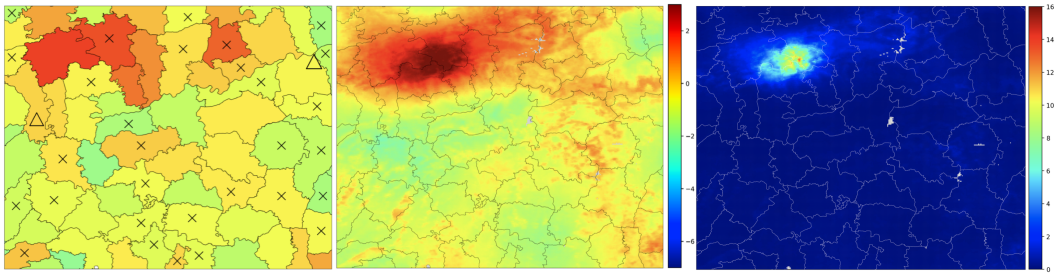

Figure 2.8: Log of malaria incidence rate λ_i^a per 1000 **Left**: Constant **Middle**: VBAgg-Exp-Obj **Right**: Square root of the variance of the Log-normal posterior on λ

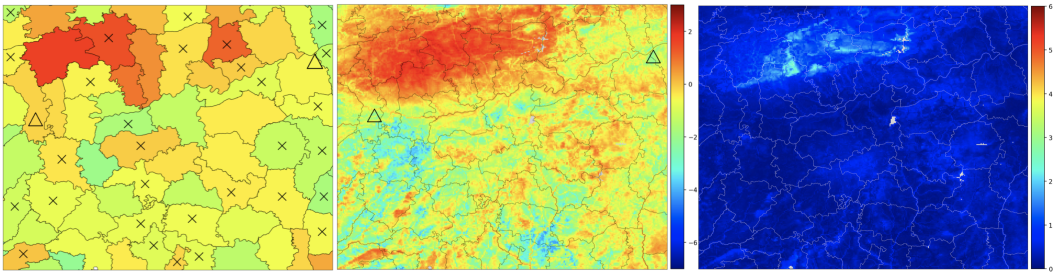

Figure 2.9: Log of malaria incidence rate λ_i^a per 1000 **Left**: Constant **Middle**: VBAgg-Sq-Obj **Right**: Square root of the variance of the non-central χ^2 posterior on λ

Although we can see that Nyström is the best performing method, the improvement over VBAgg models is not statistically significant, as shown in Table 2.4 in Appendix 2.10.4. On the other hand, both VBAgg and Nyström models statistically significantly outperform NN, which also has some instability in its predictions, as shown in the right of Figure 2.7 with further discussion to be found in Appendix 2.10.4. However, a caution should be exercised when using the measure of performance at the bag level as a surrogate for the measure of performance at the individual level: in order to perform well at the bag level, one can simply utilise spatial coordinates and ignore other covariates, as malaria intensity appears to smoothly vary between the bags (left of Figure 2.6). However, we do not expect this to be true at the individual level.

To further investigate this, we consider a particular region, and look at the predicted individual malaria incidence rate, with results found in Figure 2.7, 2.8 and 2.9 (with VBAgg tuned on Bag NLL performance to be found in Appendix 2.10.4). While Nyström and VBAgg methods both provide good bag-level performance, Nyström and VBAgg-Exp can sometimes provide overly-smooth spatial patterns, which does not seem to be the case for the VBAgg-Sq method (recall that VBAgg-Sq performed best in both prediction and calibration

for the toy experiments). To show the stability of the algorithms, we provide in Appendix 2.10.4.1 the performance across 3 different data splits, where the behaviours of each of these models can be observed. Throughout these splits, it is shown that VBAgg-Sq consistently predicts higher intensity along rivers (a known factor [Warrel et al., 2017]; indicated by triangles in Figure 2.9 as the start and end of the river) using only coarse aggregated intensities, demonstrating that prediction of (unobserved) pixel-level intensities is possible using fine-scale environmental covariates. The existence of this river is demonstrated in Figure 2.6, as rivers have higher temperature at night and it is of course more wet.

In summary, by optimising the lower bound to the marginal likelihood, the proposed variational methods are able to learn useful relations between the covariates and pixel level intensities, while avoiding the issue of overfitting to spatial coordinates. Furthermore, they also give uncertainty estimates (e.g. right of Figure 2.9), which are essential for problems like these, where validation of predictions is difficult, but they may guide policy and planning.

2.8 Reparametrisation trick for aggregate output likelihoods

This section extends the paper "Variational learning on aggregate outputs with Gaussian processes" and has not been previously published.

In a more general setting of the model discussed in this chapter, consider the aggregation function agg$(\cdot)$, which aggregates a set of GP evaluations $\{f(\mathbf{x}_i^a)\}_{i=1}^{N_a}$ for bag B_a. For example, we can take agg(B_a) to be the aggregation for the Poisson bag model:

$$\text{agg}(B_a) = \text{agg}(\mathbf{f}^a) = \sum_{i=1}^{N_a} \frac{p_i^a}{p^a} \lambda(\mathbf{x}_i^a). \tag{2.25}$$

Given now a statistical model of the form $p(y^a|\text{agg}(B_a))$, the quantity of interest is the marginal likelihood $p(\mathbf{y}|\Theta)$, where Θ corresponds to the parameters of the mean and covariance function of the GP. Using the standard ELBO derivation as reviewed in Section 1.2.2 of Chapter 1 and Section 2.4, we can obtain:

$$\sum_{a=1}^{n} \mathbb{E}_{q(\mathbf{f}^a)}[\log p\left(y^a|\text{agg}(\mathbf{f}^a)\right)] - KL\left[q\left(\mathbf{u}\right)||p\left(\mathbf{u}|W\right)\right], \tag{2.26}$$

where $\mathbf{u}$ is the set of evaluations of f at the inducing points W. With $q(\mathbf{u}) = \mathcal{N}(\boldsymbol{\eta}_{\mathbf{u}}, \boldsymbol{\Sigma}_{\mathbf{u}})$, we have the corresponding normal for $q(\mathbf{f}^a) = \int q(\mathbf{u})\, p(\mathbf{f}^a|\mathbf{u})\, d\mathbf{u}$, i.e. $q(\mathbf{f}^a) = \mathcal{N}(\mathbf{f}^a|\mathbf{m}^a, \mathbf{S}^a)$ with $\mathbf{m}^a, \mathbf{S}^a$ as defined in (2.11).

Now, because of aggregation, the first term will generally be intractable (or in the case of the Poisson bag model the first term of (2.12)) and we may opt for additional lower bound or approximation to this term which are tractable, similar to those in Section 2.4. However, since we need only unbiased estimates of its gradients with respect to variational parameters $\boldsymbol{\eta}_{\mathbf{u}}$ and $\boldsymbol{\Sigma}_{\mathbf{u}}$ as well as any model parameters Θ, e.g. parameters of the kernel function; we can simply draw $\boldsymbol{\epsilon}^a \sim \mathcal{N}(0, \mathbf{I}_{N^a})$ and use

$$\nabla_{\boldsymbol{\eta}, \boldsymbol{\Sigma}, \Theta} \log p\left(y^a | \text{agg}\left(\mathbf{m}^a + (\mathbf{S}^a)^{1/2} \boldsymbol{\epsilon}^a\right)\right).$$

as an unbiased estimator of $\nabla_{\boldsymbol{\eta}, \boldsymbol{\Sigma}, \Theta} \mathbb{E}_{q(\mathbf{f}^a)}[\log p(y^a|\text{agg}(\mathbf{f}^a))]$ (under appropriate smoothness assumptions on p, agg) [Kingma & Welling, 2013]. The advantage of using such an approach is that it allows us to define our model in more flexible ways, without introducing additional approximations whose quality may be difficult to quantify.

2.8.1 Additional experimental results

Employing this reparameterisation approach on the first term in (2.12) for the Poisson bag model, we now compare empirically the approach in Section 2.4 versus the methodology introduced here. In particular, for each mini-batch iteration per bag a we draw 100 samples from $\boldsymbol{\epsilon}^a$ to approximate the intractable integral in (2.12), before using the auto-differentiation framework of *TensorFlow* to compute unbiased gradients. Following the notation in Section 2.7, we will denote the reparameterisation methods as VBAgg-Exp-Re, VBAgg-Exp-Re-Obj, VBAgg-Sq-Re and VBAgg-Sq-Re-Obj, recalling that the Obj denotes the tuning on the training objective $\mathcal{L}$, rather than the NLL on the validation bag.

Poisson model: Swiss roll Using the same setup for the swiss roll Poisson experiment in Section 2.7.1, the results can be found in Figure 2.10 and 2.11. Here we can observe that the results are shown to be fairly similar between the two approaches, in both the *Varying number of Bags* and *Varying number of individuals per bag* experiment. However, this is not

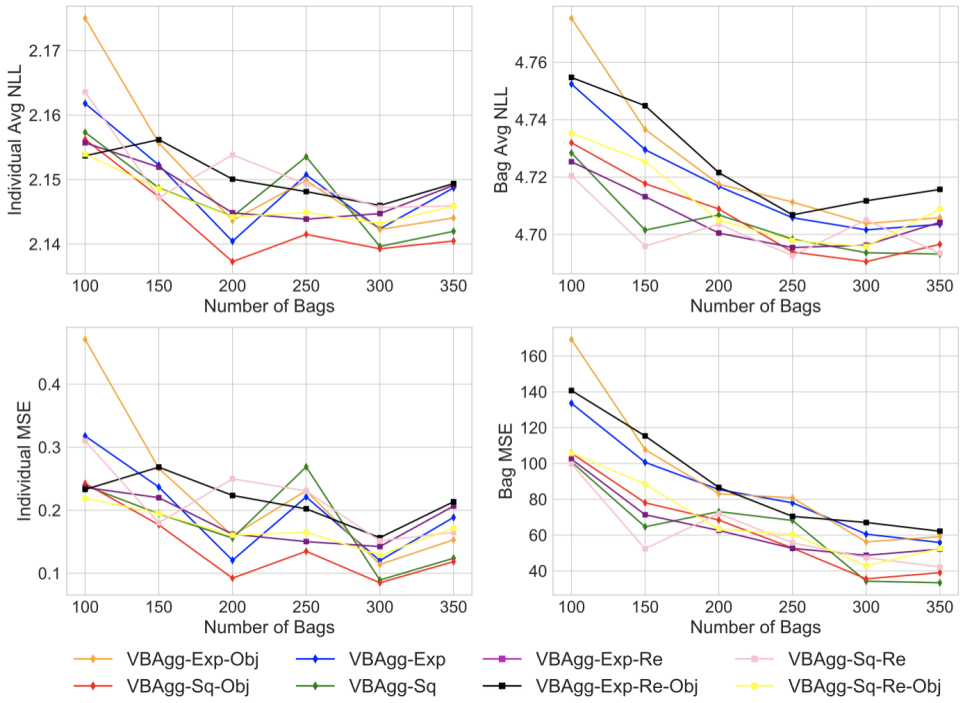

Figure 2.10: Varying number of bags over 5 repetitions. **Left Column:** Individual average NLL and MSE on train set. **Right Column:** Bag average NLL and MSE on test set (of size 500). Constant prediction NLL and MSE is 2.23 and 0.85 respectively. bag-pixel model prediction NLL is above 2.4 and MSE is above 3.0, hence not shown on graph.

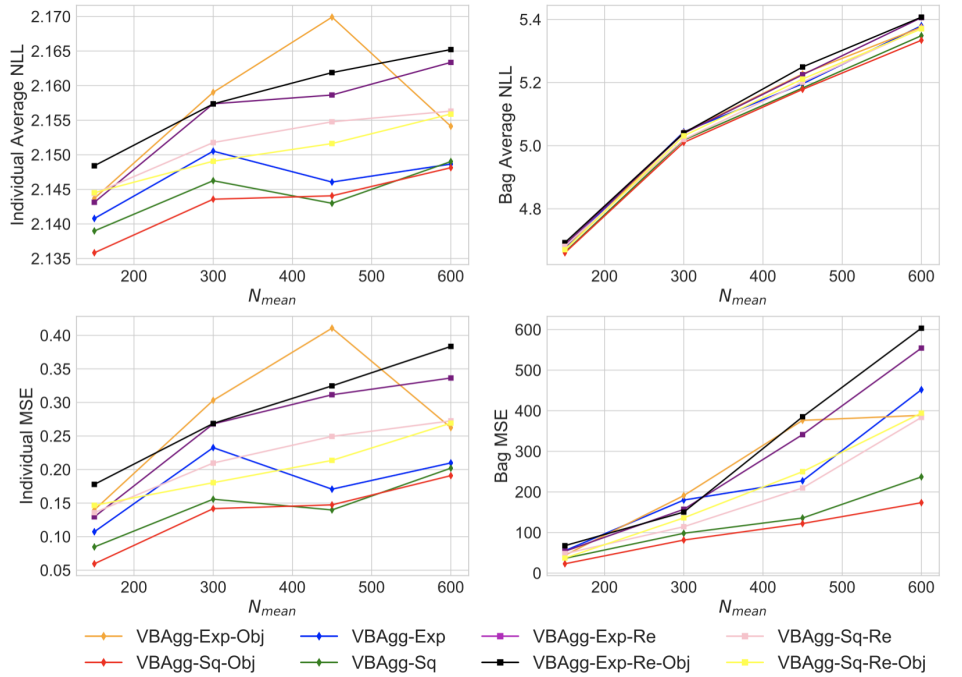

Figure 2.11: Varying number of individuals per bag N_{mean} over 5 repetitions. **Left Column:** Individual average NLL and MSE on train set. **Right Column:** Bag average NLL and MSE on test set (of size 500). Constant prediction NLL and MSE is 2.23 and 0.85 respectively.

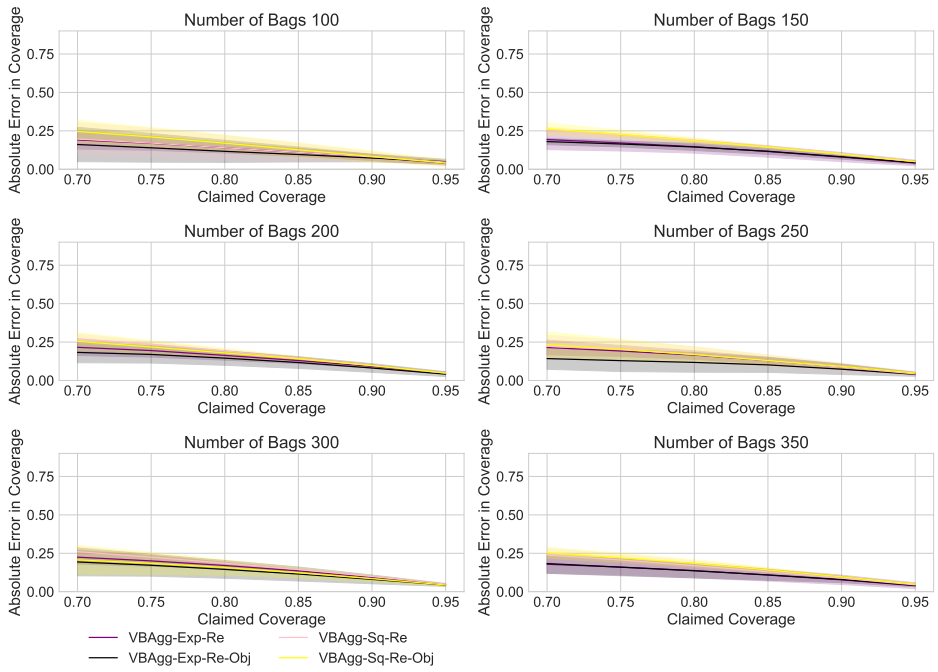

Figure 2.12: Absolute error in coverage from 70% to 95% for the increasing number of bags experiment for the Poisson model. Shaded regions highlight the standard deviation. Perfect coverage would provide a straight line at 0 error.

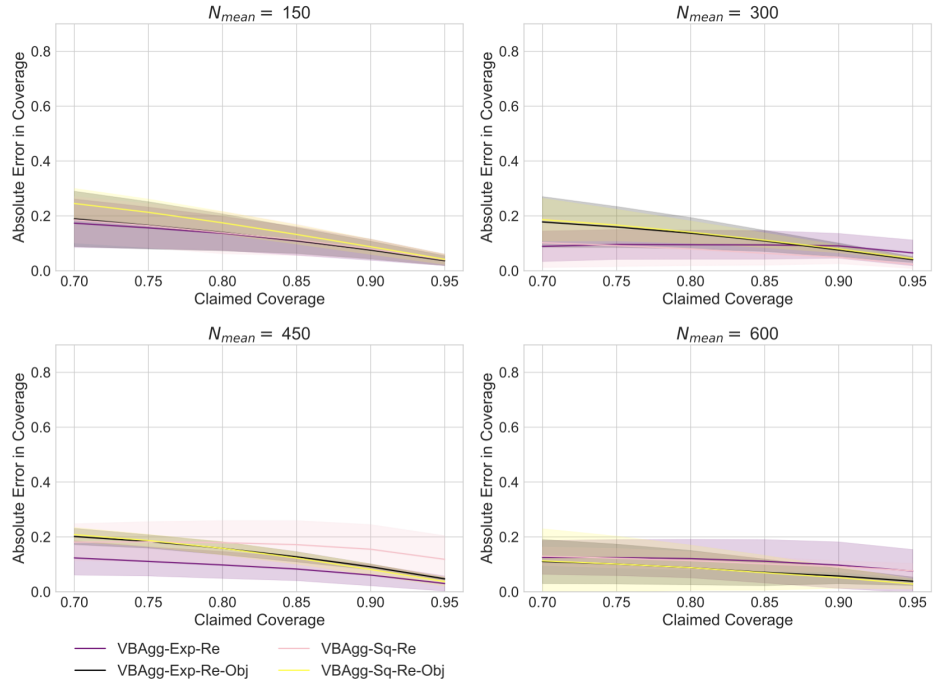

Figure 2.13: Absolute error in coverage from 70% to 95% for the increasing number of individuals per bag N_{mean} and N_{std} for the Poisson model. Shaded regions highlight the standard deviation. Perfect coverage would provide a straight line at 0 error.

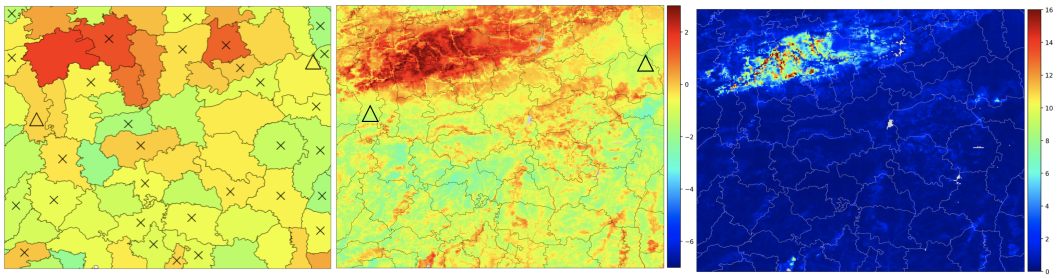

Figure 2.14: Log of malaria incidence rate λ_i^a per 1000 **Left**: Constant **Middle**: VBAgg-Exp-Re-Obj **Right**: Square root of the variance of the Log-normal posterior on λ.

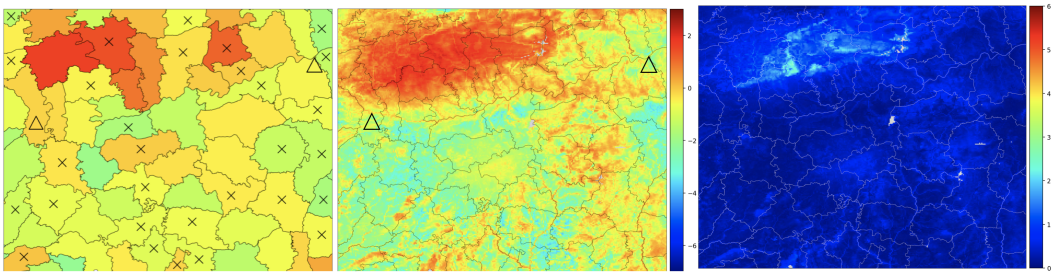

Figure 2.15: Log of malaria incidence rate λ_i^a per 1000 **Left**: Constant **Middle**: VBAgg-Sq-Re-Obj **Right**: Square root of the variance of the non-central χ^2 posterior on λ.

the case when we consider the calibration plots (Figure 2.12 and 2.13), as while VBAgg-Sq and VBAgg-Sq-Re performs similarly, VBAgg-Exp-Re has much improved calibration compared to that of Figure 2.4 and 2.5. This is not surprising, as previously VBAgg-Exp uses an extra additional lower bound on the marginal likelihood. The results here reinforce our discussion on calibration in Section 2.7.1.

Poisson model: Malaria incidence prediction Using the same experimental setup as the malaria incidence prediction in Section 2.7.3, the results for the Bag NLL and MSE on the test bags (across 10 runs) can be found in Table 2.3. Here, we observe that the reparameterisation methods which are tuned on the Bag NLL are best performing here (in terms of NLL); in particular they statistically outperform all other methods except Nyström-Exp (Table 2.6 and 2.7 in Appendix 2.10.4). It is noted this was not the case for Nyström-Exp, the best performing method previously. Focusing on the same region as before, we now investigate the behaviour of the inference methods we use here, relative to the approximation or the additional lower bound used previously. The results for VBAgg-Exp-Re-Obj and VBAgg-Sq-Re-Obj can be found in Figure 2.14 and 2.15, with its counterpart tuned on Bag NLL to be found in Figure 2.25 and 2.26 in Appendix 2.10.4. Comparing Figure 2.15 (VBAgg-Sq-Re-

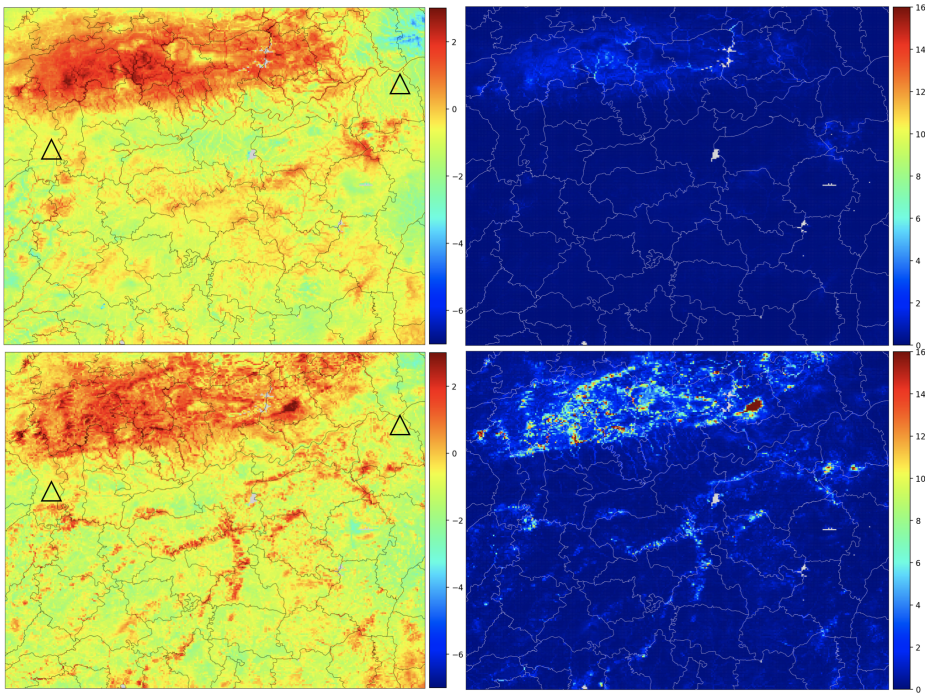

Figure 2.16: Log of malaria incidence rate λ_i^a per 1000 (left) and square root of the variance of the Log-normal posterior on λ (right) **Top**: VBAgg-Exp-Obj **Bottom**: VBAgg-Exp-Re-Obj

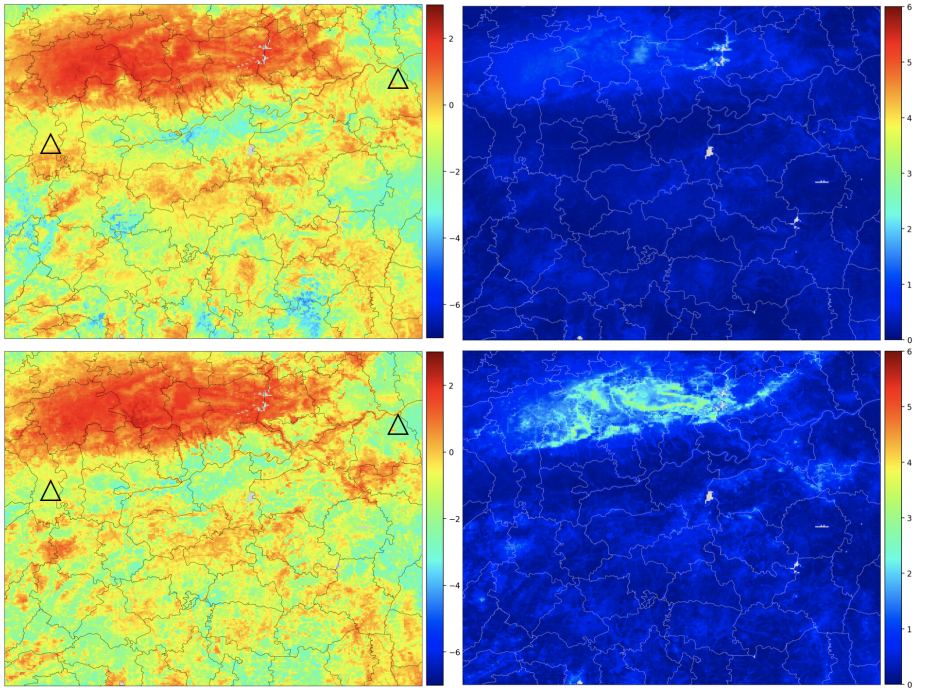

Figure 2.17: Log of malaria incidence rate λ_i^a per 1000 (left) and square root of the variance of the non-central χ^2 posterior on λ (right) **Top**: VBAgg-Sq-Obj **Bottom**: VBAgg-Sq-Re-Obj

Table 2.3: Results for the Poisson model on the malaria dataset with 10 different re-splits of the data. Bag performance is measured on a test set, with MSE computed between $\log(y^a)$ and $\log(\sum_{i=1}^{N_a} p_i^a \hat{\lambda}_i^a)$ as the underlying true rates are not known. Brackets include standard deviation. * here denotes the reparameterisation methods.

	Bag NLL	Bag MSE (Log)
Constant	173.1 (31.2)	4.08 (0.13)
VBAgg-Exp-Re*	83.8 (34.1)	1.24 (0.35)
VBAgg-Sq-Re*	87.5 (36.4)	1.32 (0.17)
VBAgg-Exp-Re-Obj*	88.3 (39.6)	1.07 (0.13)
Nyström-Exp	88.1 (25.1)	1.31 (0.15)
VBAgg-Sq-Obj	94.1 (34.0)	1.21 (0.05)
VBAgg-Exp-Obj	97.2 (39.6)	1.04 (0.11)
VBAgg-Sq	97.6 (39.0)	1.38 (0.18)
VBAgg-Sq-Re-Obj*	98.75 (44.6)	1.25 (0.15)
VBAgg-Exp	99.2 (39.8)	1.21 (0.19)
NN-Exp	164.4 (127.8)	1.82 (0.29)

Obj) to Figure 2.9 (VBAgg-Sq-Obj), we observe that the results are very similar, suggesting that the approximation that we make is very similar. It is noted that VBAgg-Exp-Re-Obj provides a much more 'realistic' smoothness of malaria incidences, unlike that of VBAgg-Exp-Obj in Figure 2.8. For a thorough comparison, we additionally display the results from using all 957 bags as training data for all the VBAgg methods (note that no validation and early stopping is needed for Obj methods). The results can be found in Figure 2.16 and 2.17. Here, across all the methods, higher intensity of malaria were predicted along some or all parts of the river (triangles denotes the start and end of the river).

2.9 Conclusion

Motivated by the vitally important problem of malaria, which is the direct cause of around 187 million clinical cases [Bhatt et al., 2015] and 631,000 deaths [Gething et al., 2016] each year in sub-Saharan Africa, we have proposed a general framework of *aggregated observation models* using Gaussian processes, along with scalable variational methods for inference in those models, making them applicable to large datasets. The proposed method allows learning in situations where outputs of interest are available at a much coarser level than that of the inputs, while explicitly quantifying uncertainty of predictions. The recent uptake of digital health information systems offers a wealth of new data which is abstracted to the

aggregate or regional levels to preserve patient anonymity. The volume of this data, as well as the availability of much more granular covariates provided by remote sensing and other geospatially tagged data sources, allows to probabilistically disaggregate outputs of interest for finer risk stratification, e.g. assisting public health agencies to plan the delivery of disease interventions. This task demands new high-performance machine learning methods and we see those that we have developed here as an important step in this direction.

2.10 Chapter appendix

2.10.1 Derivations for aggregated Exponential family models

Let $\mathbf{y} = [y^1, \ldots, y^n]$ (bag observations). With the inducing points $\mathbf{u} = f(W)$, the marginal likelihood is

$$p(\mathbf{y}) = \int \int \prod_{a=1}^{n} p(y^a|\eta^a) p(\mathbf{f}|\mathbf{u}) p(\mathbf{u}) d\mathbf{u} d\mathbf{f}. \tag{2.27}$$

The evidence lower bound can be derived as

$$\begin{aligned}
\log p(\mathbf{y}) &= \log \int \int \Big\{ \prod_{a=1}^{n} p(y^a|\eta^a) \frac{p(\mathbf{u})}{q(\mathbf{u})} \Big\} p(\mathbf{f}|\mathbf{u}) q(\mathbf{u}) d\mathbf{u} d\mathbf{f} \\
&\geq \int \int \log \Big\{ \prod_{a=1}^{n} p(y^a|\eta^a) \frac{p(\mathbf{u})}{q(\mathbf{u})} \Big\} p(\mathbf{f}|\mathbf{u}) q(\mathbf{u}) d\mathbf{u} d\mathbf{f} \\
&= \sum_{a=1}^{n} \frac{y^a}{\tau} \int F\Big(\sum_i w_i^a \Psi(f(\mathbf{x}_i^a)) \Big) q(\mathbf{f}^a) d\mathbf{f}^a - \int c\Big(F\Big(\sum_i w_i^a \Psi(f(\mathbf{x}_i^a)) \Big) \Big) q(\mathbf{f}^a) d\mathbf{f}^a \\
&\quad - \int q(\mathbf{u}) \log \frac{q(\mathbf{u})}{p(\mathbf{u})} d\mathbf{u},
\end{aligned} \tag{2.28}$$

where $q(\mathbf{f}^a) = \int p(\mathbf{f}^a|\mathbf{u}) q(\mathbf{u}) d\mathbf{u}$. By setting the variational distribution $q(\mathbf{u})$ as Gaussian, the third term is tractable. The first and second terms are however tractable only in limited cases. The cases we develop are the Poisson and normal bag model, described in the main text, as well as the Exponential bag model, described below.

For the case of the Exponential bag model, we have $F(\eta) = -1/\eta$. We can apply the similar

argument as in Lemma 2.4.1. For any $\alpha_i > 0$ with $\sum_i \alpha_i = 1$, by the concavity of F,

$$\begin{aligned}\int F\left(\sum_i w_i \Psi(v_i)\right) q(v_i) dv_i &= \int F\left(\sum_i \alpha_i w_i/\alpha_i \Psi(v_i)\right) q(v_i) dv_i \\ &\geq \int \sum_i \alpha_i F\left(w_i/\alpha_i \Psi(v_i)\right) q(v_i) dv_i \\ &= \sum_i \alpha_i \int F\left(w_i/\alpha_i \Psi(v_i)\right) q(v_i) dv_i.\end{aligned}$$

For $F(\eta) = -1/\eta$, the last line is equal to

$$\sum_i \frac{\alpha_i^2}{w_i} \int \frac{1}{\Psi(v_i)} q(v_i) dv_i.$$

When using a normal q, this is tractable for several choices of Ψ including e^v and v^2. If we let $\xi_i := \int \frac{1}{\Psi(v_i)} q(v_i) dv_i$, and maximise

$$\sum_i \alpha_i^2 \frac{\xi_i}{w_i}$$

under the constraint $\sum_i \alpha_i = 1$, we obtain

$$\alpha_i = \frac{(w_i/\xi_i)}{\sum_\ell (w_i/\xi_i)}.$$

Finally, we have a lower bound

$$\int F\left(\sum_i w^i \Psi(v_i)\right) q(v_i) dv_i \geq -\frac{\sum_i (w_i/\xi_i)}{\sum_i (w_i/\xi_i)^2} \tag{2.29}$$

where

$$\xi_i = \int \frac{1}{\Psi(v_i)} q(v_i) dv_i$$

which is tractable for a Gaussian variational family. Also with an explicit form of Ψ, it is easy to take the derivatives of the resulting lower bound with respect to the variational parameters in $q(v)$.

2.10.2 Additional details for Poisson variational derivation

2.10.2.1 Log-sum lemma

Lemma 2.10.1. *Let* $\mathbf{v} = [v_1, \ldots, v_N]^\top$ *be a random vector with probability density* $q(\mathbf{v})$*, and let* $w_i \geq 0$*,* $i = 1, \ldots, N$*. Then, for any non-negative valued function* $\Psi(v)$*,*

$$\int \log\Big(\sum_{i=1}^N w_i \Psi(v_i)\Big) q(\mathbf{v}) d\mathbf{v} \geq \log\Big(\sum_{i=1}^N w_i e^{\xi_i}\Big),$$

where

$$\xi_i := \int \log \Psi(v_i) q_i(v_i) dv_i.$$

Proof. Let $\alpha_1, \ldots, \alpha_N$ be non-negative numbers with $\sum_{i=1}^N \alpha_i = 1$. It follows from Jensen's inequality that

$$\begin{aligned}
&\int \log\big(\sum_{i=1}^N w_i \Psi(v_i)\big) q(\mathbf{v}) d\mathbf{v} \\
&= \int \log\Big(\sum_{i=1}^N \alpha_i \tfrac{w_i}{\alpha_i} \Psi(v_i)\Big) q(\mathbf{v}) d\mathbf{v} \\
&\geq \sum_{i=1}^N \alpha_i \left[\int \log\Big(\Psi(v_i)\Big) q(v_i) dv_i + \log \frac{w_i}{\alpha_i}\right] \\
&= \sum_{i=1}^N \alpha_i \xi_i + \sum_{i=1}^N \alpha_i \log \frac{w_i}{\alpha_i}. \qquad (2.30)
\end{aligned}$$

By Lagrange multiplier method, maximising the last line with respect to α gives

$$\alpha_i = \frac{w_i e^{\xi_i}}{\sum_{j=1}^N w_j e^{\xi_j}}.$$

Plugging this to (2.30) completes the proof. □

2.10.2.2 A lower bound of marginal likelihood for $\Psi(f) = e^f$ and $\Psi(f) = f^2$

Using Lemma 2.10.1, we obtain that

$$\int \log\big(\sum_{i=1}^N p_i^a \Psi(v_i^a)\big) q(\mathbf{v}^a) d\mathbf{v}^a \geq \log\Big(\sum_{i=1}^N p_i^a \Psi(\xi_i^a)\Big), \qquad (2.31)$$

where

$$\xi_i^a = \int \log \Psi(v_i^a) q_i^a(v_i^a) dv_i^a.$$

The above lower bound is tractable for the popular functions $\Psi(v) = v^2$ and $\Psi(v) = e^v$ under the normal variational distributions $q^a(\mathbf{v}^a) \sim \mathcal{N}(\mathbf{m}^a, \mathbf{S}^a)$. In particular,

$$\begin{aligned}
\Psi(v) = e^v: \quad \xi_i^a &= \int v_i^a q_i^a(v_i^a) dv_i^a = \mathbf{m}_i^a, \\
\Psi(v) = v^2: \quad \xi_i^a &= \int \log(v_i^a)^2 q_i^a(v_i^a) dv_i^a = -G\left(-\frac{\mathbf{m}_i^a}{2\mathbf{S}_{ii}^a}\right) + \log\left(\frac{\mathbf{S}_{ii}^a}{2}\right) - \gamma,
\end{aligned}$$

where γ is the Euler constant and

$$G(t) = 2t \sum_{j=0}^{\infty} \frac{j!}{(2)_j \, (3/2)_j} t^j$$

is the partial derivative of the confluent hyper-geometric function [Lloyd et al., 2015; Ancarani & Gasaneo, 2008]. However, in this work we focus on the Taylor series approximation for $\Psi(v) = v^2$, as implementation of the above bound uses a large look-up table and involves linear interpolation. Furthermore, it is suggested in experiments that the secondary lower bound proposed above in Lemma 2.10.1 can lead to poor calibration, for more details, refer to Section 2.7.

2.10.2.3 KL term

Since $q(\mathbf{u})$ and $p(\mathbf{u}|W)$ are both normal distribution, the KL divergence is tractable:

$$KL(q(\mathbf{u})||p(\mathbf{u}|W)) = \frac{1}{2}\Big\{tr[\mathbf{K}_{WW}^{-1}\boldsymbol{\Sigma}_{\mathbf{u}}] + \log\frac{|\mathbf{K}_{WW}|}{|\boldsymbol{\Sigma}_{\mathbf{u}}|} - m + (\boldsymbol{\mu}_W - \boldsymbol{\eta}_{\mathbf{u}})^T \mathbf{K}_{WW}^{-1}(\boldsymbol{\mu}_W - \boldsymbol{\eta}_{\mathbf{u}})\Big\}. \tag{2.32}$$

2.10.2.4 Taylor series approximation in the variational method

We consider the integral

$$\int \log\big(\sum_{i=1}^{N} p_i^a (v_i^a)^2\big) q^a(\mathbf{v}^a) d\mathbf{v}^a$$

where q^a is $\mathcal{N}(\mathbf{m}^a, \mathbf{S}^a)$. Note that this can be written as $\mathbb{E}[\log \|\tilde{\mathbf{v}}^a\|^2]$, where $\tilde{\mathbf{v}}^a \sim N(\tilde{\mathbf{m}}^a, \tilde{\mathbf{S}}^a)$ with $\mathbf{P}^a = \text{diag}\left(p_1^a, \ldots, p_{N_a}^a\right)$, $\tilde{\mathbf{m}}^a = (\mathbf{P}^a)^{1/2}\mathbf{m}^a$ and $\tilde{\mathbf{S}}^a = (\mathbf{P}^a)^{1/2}\mathbf{S}^a(\mathbf{P}^a)^{1/2}$. Note that $\|\tilde{\mathbf{v}}^a\|^2$ follows a non-central chi-squared distribution. We now resort to a Taylor series approximation for $\mathbb{E}[\log \|\tilde{\mathbf{v}}^a\|^2]$ (similar to Teh et al. [2007]) around $\mathbb{E}[\|\tilde{\mathbf{v}}^a\|^2] = \|\tilde{\mathbf{m}}^a\|^2 + tr(\tilde{\mathbf{S}}^a)$, resulting in

$$\begin{aligned}
\mathbb{E}[\log\left(\|\tilde{\mathbf{v}}^a\|^2\right)] &= \log\left(\mathbb{E}[\|\tilde{\mathbf{v}}^a\|^2]\right) \\
&\quad + \mathbb{E}\left[\frac{\|\tilde{\mathbf{v}}^a\|^2 - \mathbb{E}[\|\tilde{\mathbf{v}}^a\|^2]}{\mathbb{E}[\|\tilde{\mathbf{v}}^a\|^2]} - \frac{\left(\|\tilde{\mathbf{v}}^a\|^2 - \mathbb{E}[\|\tilde{\mathbf{v}}^a\|^2]\right)^2}{2\left(\mathbb{E}[\|\tilde{\mathbf{v}}^a\|^2]\right)^2} + \mathcal{O}\left(\left(\|\tilde{\mathbf{v}}^a\|^2 - \mathbb{E}[\|\tilde{\mathbf{v}}^a\|^2]\right)^3\right)\right] \\
&\approx \log\left(\|\tilde{\mathbf{m}}^a\|^2 + tr(\tilde{\mathbf{S}}^a)\right) - \frac{2\tilde{\mathbf{m}}^{a\top}\tilde{\mathbf{S}}^a\tilde{\mathbf{m}}^a + tr\left(\left(\tilde{\mathbf{S}}^a\right)^2\right)}{\left(\|\tilde{\mathbf{m}}^a\|^2 + tr(\tilde{\mathbf{S}}^a)\right)^2}.
\end{aligned}$$

As commented in Teh et al. [2007], approximation is very accurate when $\mathbb{E}[\|\tilde{\mathbf{v}}^a\|^2]$ is large, but the caveat is that the Taylor series converges only for $\|\tilde{\mathbf{v}}^a\|^2 \in (0, 2\mathbb{E}[\|\tilde{\mathbf{v}}^a\|^2])$ so this approach effectively ignores the tail of the non-central chi-squared.

2.10.3 Additional information for baselines

2.10.3.1 Random Fourier features on Laplacian

Here we discuss using random Fourier features [Rahimi & Recht, 2007] to reduce computational cost in calculation of the Laplacian defined as $\mathbf{L} = \text{diag}(\mathbf{K}\mathbf{1}^\top) - \mathbf{K}$, where $\mathbf{1}$ is just $[1, \ldots, 1]$ and $\mathbf{K}$. Suppose the kernel is stationary i.e. $k_w(\mathbf{x} - \mathbf{y}) = k(\mathbf{x}, \mathbf{y})$ (some examples include the Gaussian and Matérn kernel), then using random Fourier features, we obtain $\mathbf{K} \approx \mathbf{\Phi}\mathbf{\Phi}^\top$, where $\mathbf{\Phi} \in \mathbb{R}^{b_N \times m}$, b_N denotes the total number of individuals in the batch and m denotes the number of frequencies. Now we have:

$$\mathbf{f}^\top \mathbf{L}\mathbf{f} \approx \mathbf{f}^\top \text{diag}(\mathbf{\Phi}\mathbf{\Phi}^\top \mathbf{1}^\top)\mathbf{f} - \mathbf{f}^\top \mathbf{\Phi}\mathbf{\Phi}^\top \mathbf{f} = \mathbf{f}^\top \text{diag}(\mathbf{\Phi}\mathbf{\Phi}^\top \mathbf{1}^\top)\mathbf{f} - ||\mathbf{\Phi}^\top \mathbf{f}||_2^2 \tag{2.33}$$

In both terms, we can avoid computing the kernel matrix, by carefully selecting the order of computation. Note another option is to consider Nyström approximation with landmark points $\{\mathbf{w}_1, \ldots \mathbf{w}_h\}$, then $\mathbf{K} \approx \mathbf{K}_{XW}\mathbf{K}_{WW}^{-1}\mathbf{K}_{WX}$, where $\mathbf{K}_{WW}$ denotes the kernel matrix on landmark points, while $\mathbf{K}_{XW}$ is the kernel matrix between landmark points and the data $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. Then $\mathbf{\Phi} = \mathbf{K}_{XW}\mathbf{K}_{WW}^{-\frac{1}{2}}$.

2.10.3.2 Bag manifold regularisation

Suppose we have bag covariates $\mathbf{s}^a$ (note these are for the entire bag), and also some summary statistics of a bag, e.g. mean embeddings [Muandet et al., 2017] given by $\mathbf{h}^a_{\text{embed}} = \frac{1}{N_a}\sum_{i=1}^{N_a} h(\mathbf{x}_i^a)$, with some user-defined h. Then similar to individual level manifold regularisation, we can consider manifold regularisation at the bag level (assuming a separable kernel for simplicity), i.e.

$$\ell_2 = \sum_{l=1}^{n}\sum_{m=1}^{n}(f^l_{\text{embed}} - f^m_{\text{embed}})^2 k_s(\mathbf{s}^l, \mathbf{s}^m) k_h(\mathbf{h}^l_{\text{embed}}, \mathbf{h}^m_{\text{embed}}) = \mathbf{f}^\top_{\text{embed}} \mathbf{L}_{\text{bag}} \mathbf{f}_{\text{embed}} \tag{2.34}$$

where $f^a_{\text{embed}} = \frac{1}{N_l}\sum_{i=1}^{N_a} f^a_i$, k_s is a kernel on bag covariates $\mathbf{s}^a$, k_μ is a kernel on $\mathbf{h}^a_{\text{embed}}$, $\mathbf{L}_{\text{bag}}$ is the bag level Laplacian with the corresponding kernel, and $\mathbf{f}_{\text{embed}} = [f^1_{\text{embed}}, \ldots, f^n_{\text{embed}}]^\top$. Combining all these terms, we have the following loss function to minimise:

$$\ell = \frac{1}{b}\ell_0 + \frac{\lambda_1}{b_N^2}\ell_1 + \frac{\lambda_2}{b_N^2}\ell_2 \tag{2.35}$$

where b is the mini-batch size in SGD, b_N is the total number of individuals in each mini-batch, λ_1 and λ_2 are parameters controlling the strength of the respective regularisation.

2.10.4 Additional Malaria experimental results

Here we provide additional experimental results for the malaria dataset. Statistical significance (for Table 2.2) was not establish for the best performing Nyström method versus the VBAgg methods, this is shown in Table 2.4. We further provide additional prediction/uncertainty patches for 3 different splits to highlight the general behaviour of the trained models, with further explanation and details below. It is also noted in all cases λ^a_i is the incidence rate per 1000 people. Here, we learn any scale parameters and weights during training. For the NN, we also use this kernel as part of the manifold regularisation, however we use a RBF kernel instead of an ARD kernel, due to parameter tuning reasons (we can no longer learn these scales).

For constant model, bag rate predictions are computed by, $p^a\hat{\lambda}^{\text{bag}}_c$,where

$$\hat{\lambda}^{\text{bag}}_c = \frac{1}{\sum_{a=1}^n p^a}\sum_{a=1}^n y^a.$$

This essentially takes into account of population.

Table 2.4: p-values from a Wilcoxon signed-rank test for Nyström-Exp versus the methods below for Bag NLL and MSE for the malaria dataset. The null hypothesis is Nyström-Exp performs equal or worse than the considered method on the test bag performance.

	NLL	MSE
Constant	0.0009766	0.0009766
NN-Exp	0.00293	0.0009766
VBAgg-Sq-Obj	0.1162	0.958
VBAgg-Sq	0.1377	0.1611
VBAgg-Exp-Obj	0.08008	1.0
VBAgg-Exp	0.09668	0.958

Table 2.5: p-values from a Wilcoxon signed-rank test for VBAgg-Sq versus the methods below for Bag NLL and MSE for the malaria dataset. The null hypothesis is VBAgg-Sq performs equal or worse than the considered method on the test bag performance.

	NLL	MSE
Constant	0.0009766	0.0009766
NN-Exp	0.01855	0.001953
VBAgg-Sq-Obj	0.6234	0.9861
Nyström-Exp	0.8838	0.8623
VBAgg-Exp-Obj	0.6875	1.0
VBAgg-Exp	0.3477	0.9346

Table 2.6: p-values from a Wilcoxon signed-rank test for VBAgg-Exp-Re versus the methods below for Bag NLL and MSE for the malaria dataset. The null hypothesis is VBAgg-Exp-Re performs equal or worse than the considered method on the test bag performance.

	NLL	MSE
Constant	0.0009766	0.0009766
NN-Exp	0.006836	0.0009766
Nyström-Exp	0.1611	0.2158
VBAgg-Sq-Obj	0.02441	0.6152
VBAgg-Sq	0.00293	0.1611
VBAgg-Exp-Obj	0.0009766	0.9678
VBAgg-Exp	0.02441	0.6152

Table 2.7: p-values from a Wilcoxon signed-rank test for VBAgg-Sq-Re versus the methods below for Bag NLL and MSE for the malaria dataset. The null hypothesis is VBAgg-Sq-Re performs equal or worse than the considered method on the test bag performance.

	NLL	MSE
Constant	0.0009766	0.0009766
NN-Exp	0.006836	0.0009766
Nyström-Exp	0.3848	0.4229
VBAgg-Sq-Obj	0.009766	0.8838
VBAgg-Sq	0.004883	0.1875
VBAgg-Exp-Obj	0.04199	0.998
VBAgg-Exp	0.04199	0.8389

2.10.4.1 Predicted log malaria incidence rate for various models

Constant: Bag level observed incidences This is the baseline with $\hat{\lambda}_i^a$ being constant throughout the bag, as shown in Figure 2.18. For training, we only use 60% of the data.

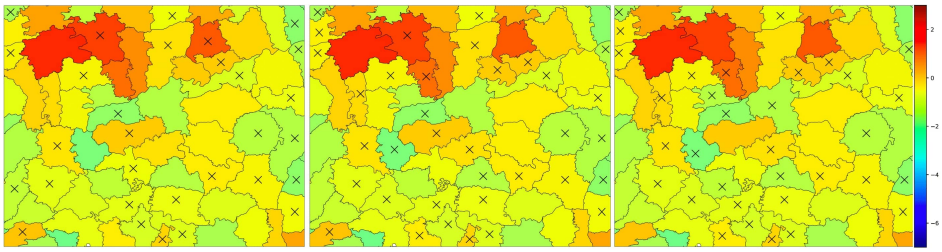

Figure 2.18: Predicted $\hat{\lambda}_i^a$ on log scale using constant model, for 3 different re-splits of the data.$\times$ denote non-train set bags.

VBAgg-Sq-Obj This is the VBAgg model with $\Psi(v) = v^2$ and tuning of hyperparameters is performed based on training objective, the lower bound to the marginal likelihood. It is noted that here we ignore early-stop and validation set here. Malaria incidence was predicted to be higher near the river, as discussed in Section 2.7.3.

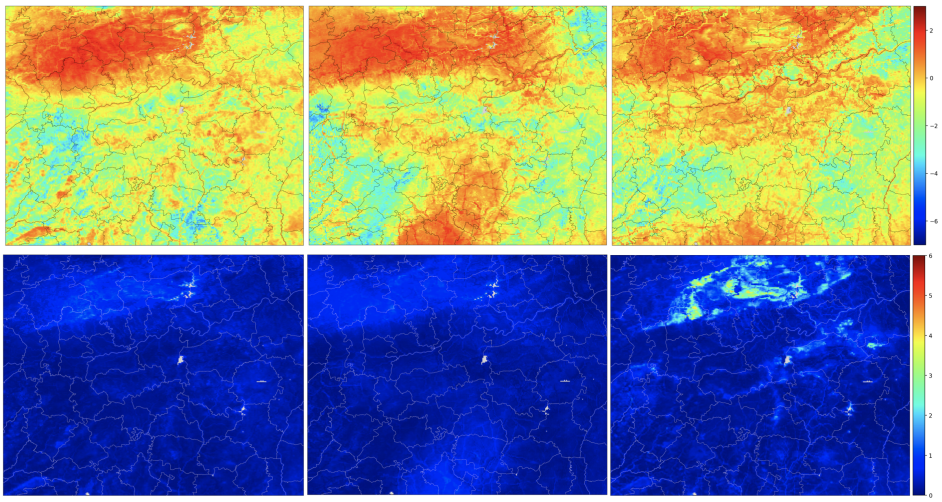

Figure 2.19: **Top:** Predicted $\hat{\lambda}_i^a$ per 1000 people on log scale for VBAgg-Sq-Obj. **Bottom:** Square root of the variance of the non-central χ^2 posterior on λ.

VBAgg-Sq This is the VBAgg model with $\Psi(v) = v^2$ and tuning of hyperparameters is performed based on NLL at the bag level. Predicted incidence are similar to the VBAgg-Sq-Obj model. In the first patch, the same parameters was chosen as VBAgg-Sq-Obj.

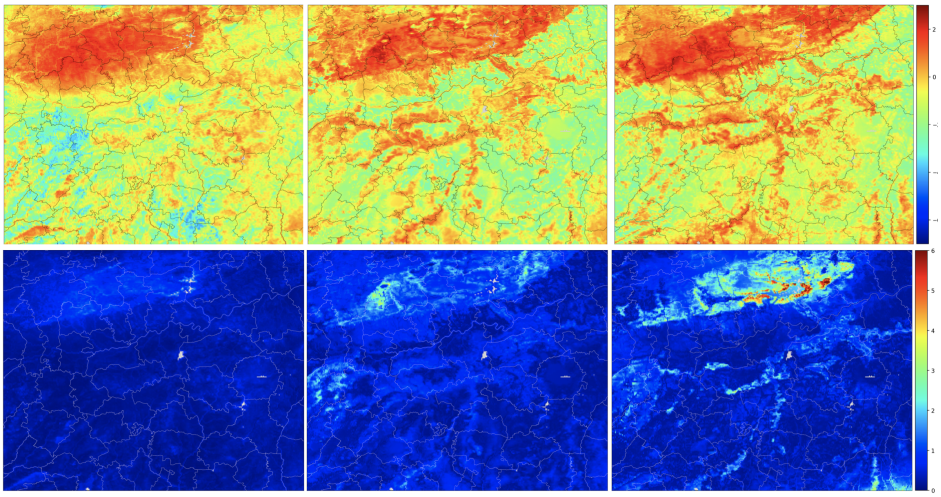

Figure 2.20: **Top:** Predicted $\hat{\lambda}_i^a$ per 1000 people on log scale for VBAgg-Sq. **Bottom:** Square root of the variance of the non-central χ^2 posterior on λ.

VBAgg-Exp-Obj This is the VBAgg model with $\Psi(v) = e^v$ and tuning of hyperparameters is performed based on training objective, the lower bound to the marginal likelihood, we ignore the early-stop and validation set here. Predicted incidence seem to be stable in general, though some smoothness is observed.

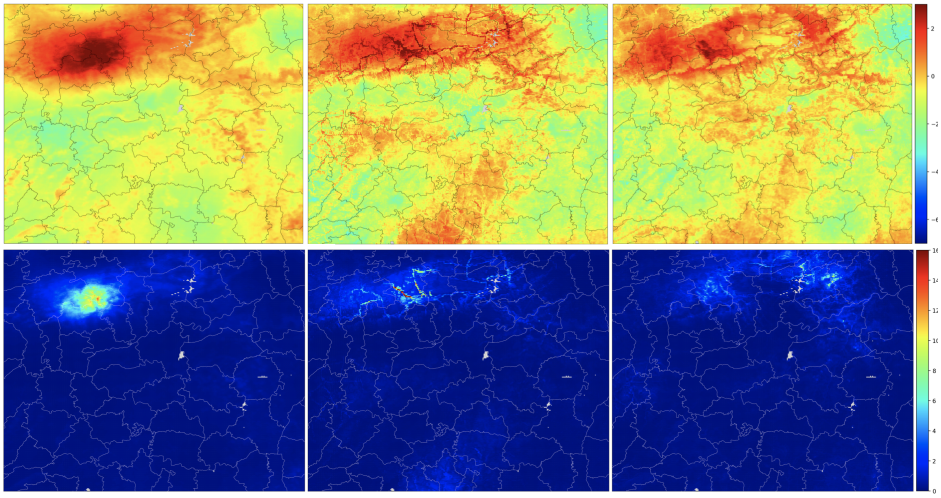

Figure 2.21: **Top:** Predicted $\hat{\lambda}_i^a$ per 1000 people on log scale for VBAgg-Exp-Obj. **Bottom:** Square root of the variance of the Log-normal posterior on λ.

VBAgg-Exp This is the VBAgg model with $\Psi(v) = e^v$ and tuning of hyperparameters is performed based on NLL.

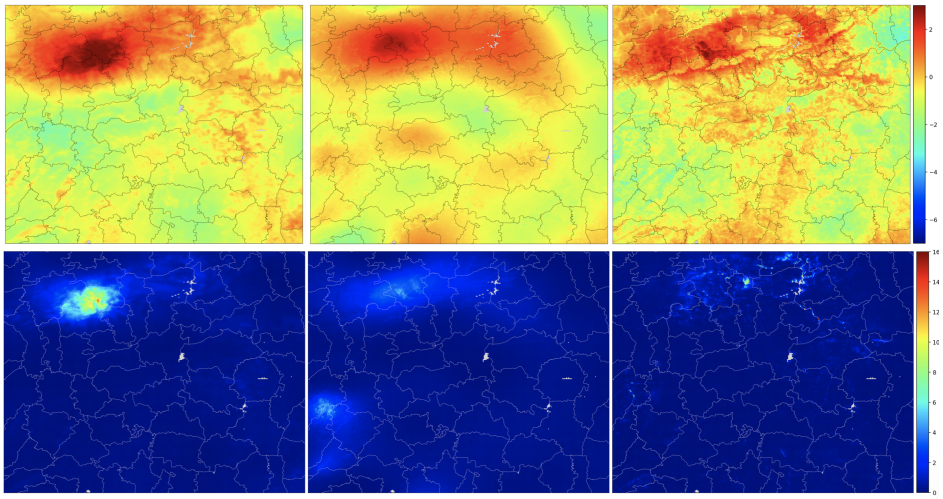

Figure 2.22: **Top:** Predicted $\hat{\lambda}_i^a$ per 1000 people on log scale for VBAgg-Exp. **Bottom:** Square root of the variance of the Log-normal posterior on λ.

Nyström-Exp This is the Nyström-Exp model, it is clear that while it performs best in terms of bag NLL, sometimes prediction are too smooth in the pixel space, this is because it optimises directly bag NLL. This pattern might be seen to be unrealistic, and may cause useful covariates to be neglected.

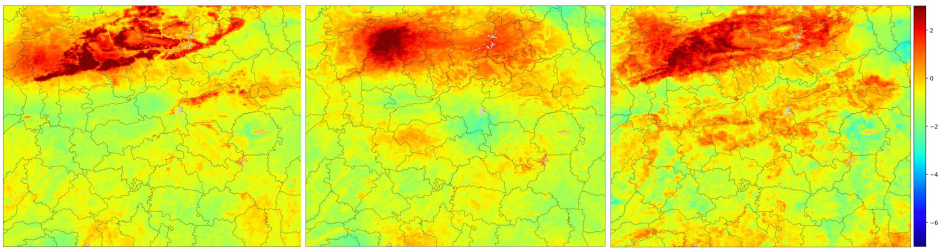

Figure 2.23: Predicted $\hat{\lambda}_i^a$ per 1000 people on log scale for Nyström-Exp.

NN-Exp We can see that the model is not very stable, and this is likely due to the model does not have an inbuilt spatial smoothness function unlike other methods. Also, the maximum predicted pixel level intensity rate $\hat{\lambda}_i^a$ is over 1000 in some cases, which is impossible.

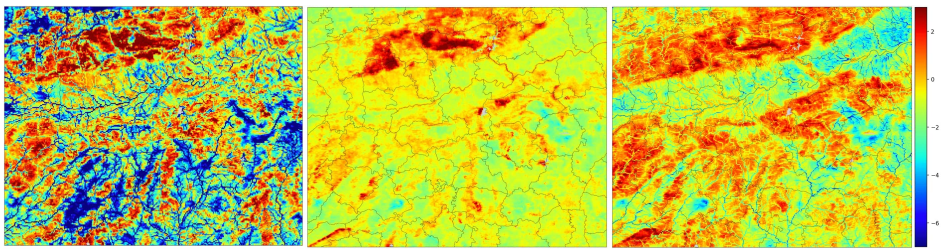

Figure 2.24: Predicted $\hat{\lambda}_i^a$ per 1000 people on log scale for NN-Exp.

VBAgg-Exp-Re Additional results for the VBAgg-Exp-Re method.

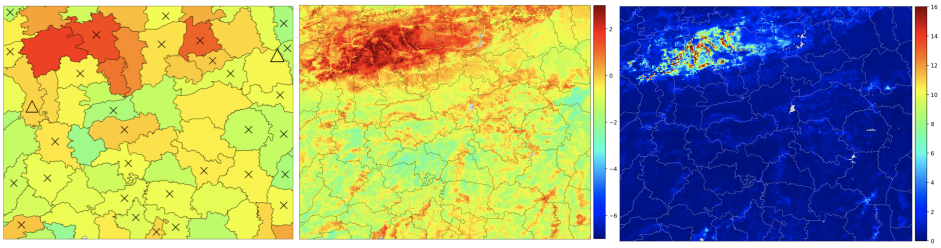

Figure 2.25: Log of malaria incidence rate λ_i^a per 1000 **Left**: Constant **Middle**: VBAgg-Exp-Re **Right**: Square root of the variance of the Log-normal posterior on λ.

VBAgg-Sq-Re Additional results for the VBAgg-Sq-Re method.

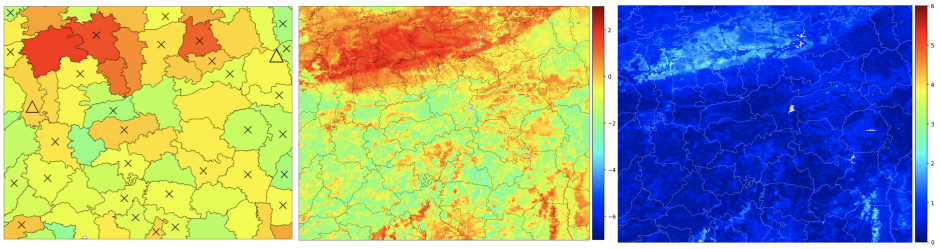

Figure 2.26: Log of malaria incidence rate λ_i^a per 1000 **Left**: Constant **Middle**: VBAgg-Sq-Re **Right**: Square root of the variance of the non-central χ^2 posterior on λ.

2.10.5 Additional Toy experimental results

In this section, we provide additional experimental results for the normal and Poisson model.

For calibration plots, we compute the α quantiles of the approximated posterior distribution and consider the ratio of times the underlying rate parameter λ_i^a (or μ_i^a for the normal model) appear inside the quantiles of the posterior distribution. If the model provides good uncertainties/calibration, we should expect to see the quantiles to match with the observed ratio. Calibration plots can be found in Figure 2.31 and Figure 2.32 for the normal model, with Figure 2.4 and Figure 2.5 for the Poisson model.

To demonstrate statistical significance of our result, we aggregate the repetitions in each experiment for each method and consider a one sided rank permutation test (Wilcoxon signed-rank test) to see whether VBAgg is statistically significant better than other approaches for individual NLL and MSE.

2.10.5.1 Poisson model

The varying number of bags experimental results is found in Figure 2.2, with the corresponding table of p-values in Table 2.8, 2.9 demonstrating statistical significance of the VBAgg-Exp and VBAgg-Sq method. Similarly, the varying number of individuals per bag through N_{mean} experimental result can be found in Figure 2.3, with the corresponding table of p-values in Table 2.10, 2.11. The comparison between VBAgg-Exp and VBAgg-Sq was found to be non-significant.

Table 2.8: p-values from a Wilcoxon signed-rank test for VBAgg-Sq versus the methods below for the varying number of bags experiment for the Poisson model. The null hypothesis is VBAgg-Sq performs equal or worse than NN or Nyström in terms of individual NLL or MSE on the train set.

	NLL	MSE
NN-Exp	$6.98\text{e}{-06}$	0.00025
Nyström-Exp	0.00048	0.00015

Table 2.9: p-values from a Wilcoxon signed-rank test for VBAgg-Exp versus the methods below for the varying number of bags experiment for the Poisson model. The null hypothesis is VBAgg-Exp performs equal or worse than NN or Nyström in terms of individual NLL or MSE on the train set.

	NLL	MSE
NN-Exp	$2.48\text{e}{-06}$	$2.48\text{e}{-05}$
Nyström-Exp	0.0005	0.00025

Table 2.10: p-values from a Wilcoxon signed-rank test for VBAgg-Sq versus the methods below for the varying number of individuals per bag experiment for the Poisson model. The null hypothesis is VBAgg-Sq performs equal or worse than NN or Nyström in terms of individual NLL or MSE on the train set.

	NLL	MSE
NN-Exp	$1.81\text{e}{-05}$	$9.53\text{e}{-06}$
Nyström-Exp	0.062	0.041

Table 2.11: p-values from a Wilcoxon signed-rank test for VBAgg-Exp versus the methods below for the varying number of individuals per bag experiment for the Poisson model. The null hypothesis is VBAgg-Exp performs worse than NN or Nyström in terms of individual NLL or MSE on the train set.

	NLL	MSE
NN-Exp	$6.68\text{e}{-05}$	0.00016
Nyström-Exp	0.049	0.062

Prediction and uncertainty plots In Figure 2.27 and 2.28, we provide some prediction plots for different models, and uncertainties for VBAgg models.

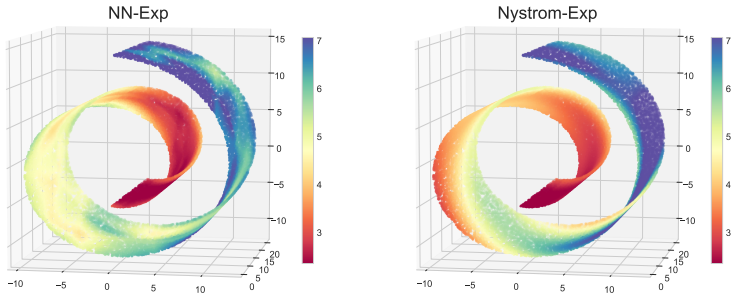

Figure 2.27: Individual predictions on the train set for the swiss roll dataset with 150 bags for NN and Nyström model. Here $N_{mean} = 150$, with $N_{std} = 50$.

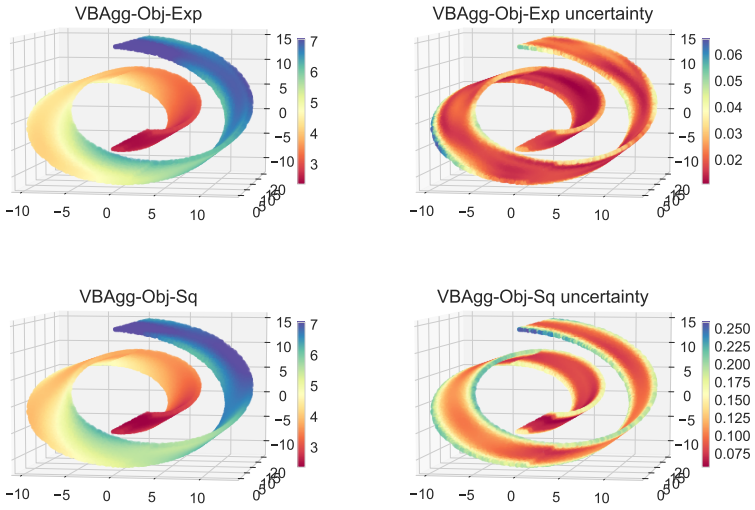

Figure 2.28: Predictions and uncertainty on the swiss roll dataset with 150 bags for the VBAgg-Obj models. Here $N_{mean} = 150$, with $N_{std} = 50$. For uncertainty, we plot the standard deviation of the posterior of $\mathbf{v}$, coming from $\mathbf{v}^a \sim \mathcal{N}(\mathbf{m}^a, \mathbf{S}^a)$ in (2.11).

2.10.5.2 Normal model

In this section, we provide some experimental results for the normal model, where throughout we assume $\tau_i^a = \tau$, same for all individuals. We consider the same swiss roll dataset as in the Poisson model and take the colour of each point to be the underlying mean μ_i^a. We then consider $y_i^a \sim \mathcal{N}(\mu_i^a, \tau)$ with $\tau = 0.1$, hence bag observations are given by

$y^a = \sum_{i=1}^{N_a} y_i^a \sim \mathcal{N}(\mu^a, N_a\tau)$ with $\mu^a = \sum_{i=1}^{N_a} \mu_i^a$. Here, the goal is to predict μ_i^a and τ, given bag observations y^a only. The results for the experiments are shown below in Figure 2.29 and Figure 2.30, which shows the VBAgg outperforming the NN and Nyström model. To show statistical significance, we also report the corresponding table of p-values in Table 2.12 and Table 2.13. Furthermore, we would also like to point out that the VBAgg is well calibrated as shown in Figure 2.31.

In Figure 2.31 and 2.32, we provide calibration results for both experiments that we have considered. It is clear that VBAgg-Obj has better calibration in general, this is not surprising as it is tuned based on the correct objective, rather than NLL. We also provide additional prediction and uncertainty plots in Figure 2.33 and 2.34.

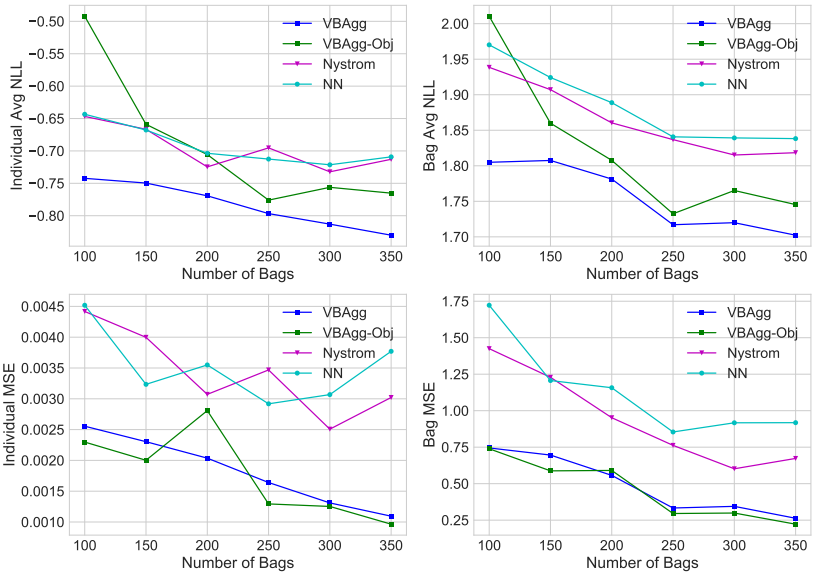

Figure 2.29: Varying number of bags over 5 repetitions for the normal model.**Left Column:** Individual average NLL and MSE on train set. **Right Column:** Bag average NLL and MSE on test set (of size 500). Constant model individual MSE is 0.04.

Table 2.12: p-values from a Wilcoxon signed-rank test for VBAgg versus the methods below for the varying number of bags experiment for the normal model. The null hypothesis is VBAgg performs equal or worse than NN or Nyström in terms of individual NLL or MSE on the train set.

	NLL	MSE
NN	5.96e−07	4.79e−09
Nyström	4.01e−08	6.52e−09

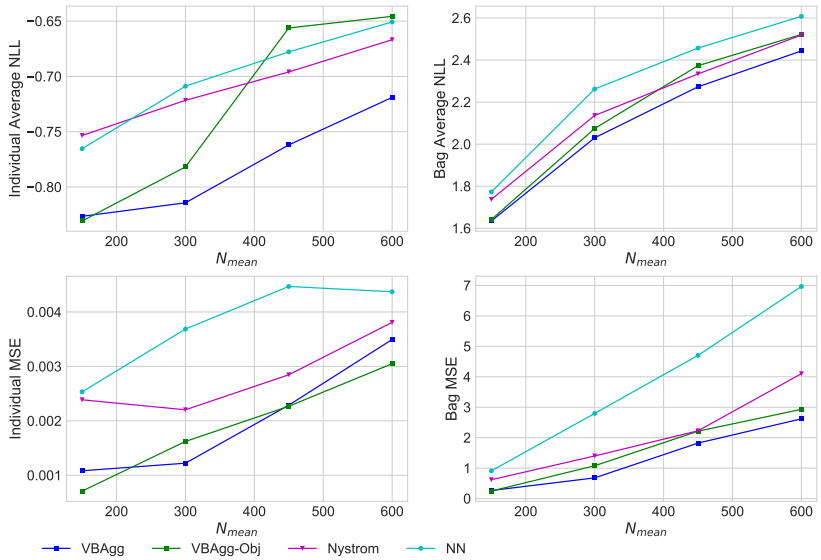

Figure 2.30: Varying number of individuals per bag N_{mean} over 5 repetitions. **Left Column:** Individual average NLL and MSE on train set. **Right Column:** Bag average NLL and MSE on test set (of size 500). Constant model individual MSE is 0.039.

Table 2.13: p-values from a Wilcoxon signed-rank test for VBAgg versus the methods below for the varying number of individuals per bag N_{mean} experiment for the normal nodel. The null hypothesis is VBAgg performs worse than NN or Nyström in terms of individual NLL or MSE on the train set.

	NLL	MSE
NN	4.77e−06	4.77e−06
Nyström	4.77e−06	4.77e−06

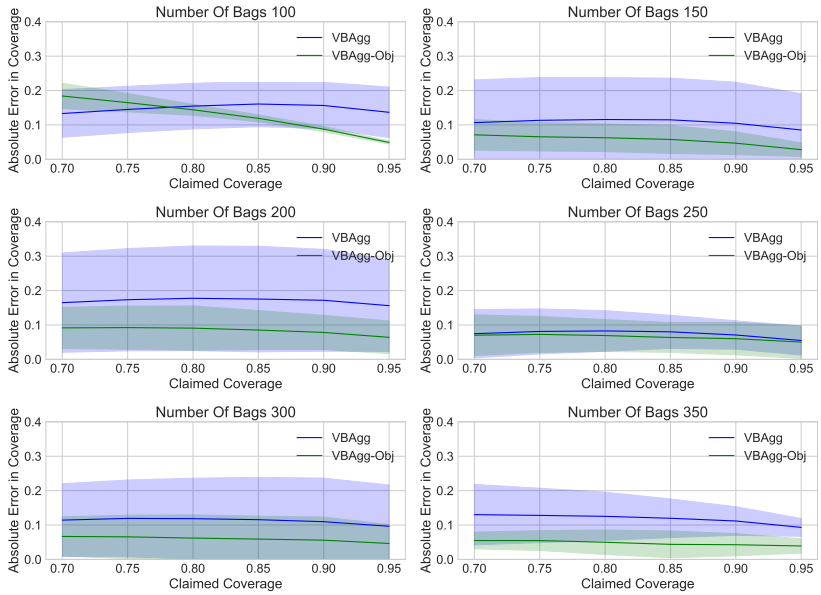

Figure 2.31: Absolute error in coverage from 70% to 95% for the increasing number of bags experiment for the normal model. Shaded regions highlight the standard deviation. Perfect coverage would provide a straight line at 0 error.

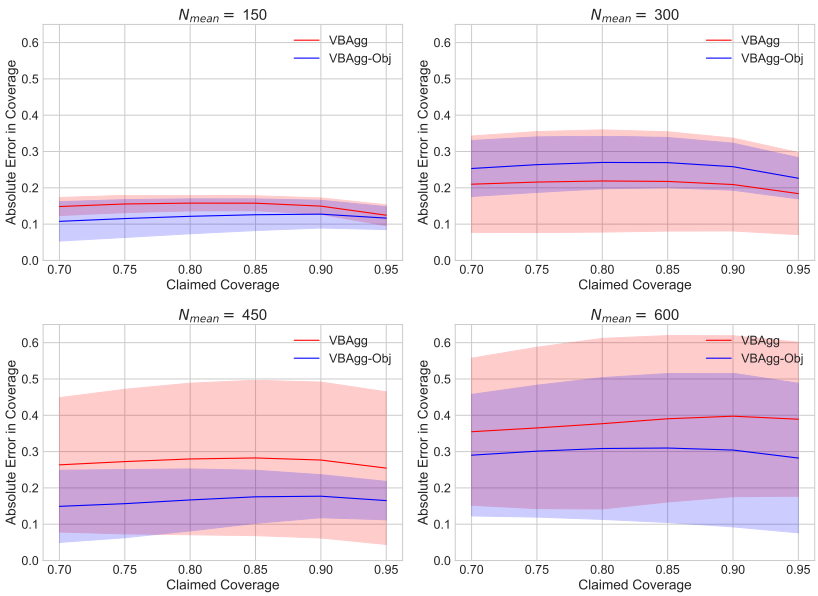

Figure 2.32: Absolute error in coverage from 70% to 95% for the increasing number of individuals per bag N_{mean} and N_{std} for the normal model. Shaded regions highlight the standard deviation. Perfect coverage would provide a straight line at 0 error.

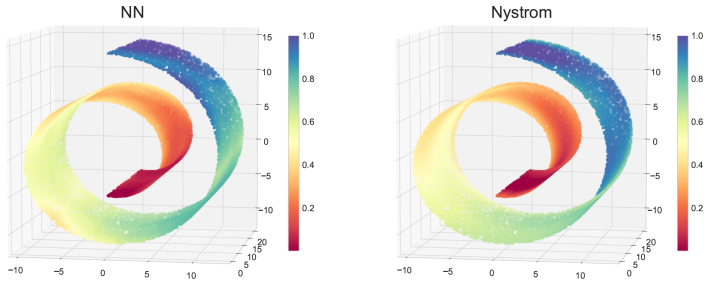

Figure 2.33: Individual predictions on the train set for the swiss roll dataset with 150 bags for NN and Nyström model. Here $N_{mean} = 150$, with $N_{std} = 50$.

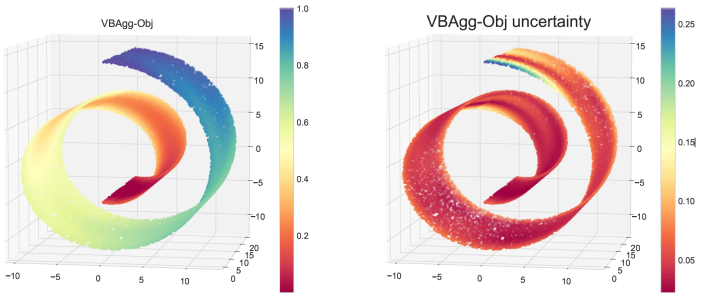

Figure 2.34: Predictions and uncertainty on the swiss roll dataset with 150 bags for the VBAgg-Obj model. Here $N_{mean} = 150$, with $N_{std} = 50$. For uncertainty, we plot the standard deviation of the posterior of $\mathbf{v}$, coming from $\mathbf{v}^a \sim \mathcal{N}(\mathbf{m}^a, \mathbf{S}^a)$ in (2.11).

Chapter 3

Hyperparameter Learning via Distributional Transfer

This chapter is based on the following paper:

Ho Chung Leon Law, Peilin Zhao, Lucian Chan, Junzhou Huang, and Dino Sejdinovic
Hyperparameter Learning via Distributional Transfer [Law et al., 2018c]
Advances in Neural Information Processing Systems (NeurIPS), 2019

Bayesian optimisation is a popular technique for hyperparameter learning but typically requires initial exploration even in cases where similar prior tasks have been solved. We propose to transfer information across tasks using learnt representations of training datasets used in those tasks. This results in a joint Gaussian process model on hyperparameters and data representations. Representations make use of the framework of distribution embeddings into reproducing kernel Hilbert spaces. The developed method has a faster convergence compared to existing baselines, in some cases requiring only a few evaluations of the target objective.

3.1 Introduction

Hyperparameter selection is an essential part of training a machine learning model and a judicious choice of values of hyperparameters such as learning rate, regularisation, or kernel parameters is what often makes the difference between an effective and a useless model. To tackle the challenge in a more principled way, the machine learning community has been increasingly focusing on Bayesian optimisation (BO) [Snoek et al., 2012], a sequential strategy

to select hyperparameters $\boldsymbol{\theta}$ based on past evaluations of model performance. In particular, a Gaussian process (GP) [Rasmussen & Williams, 2006] prior is used to represent the underlying accuracy f as a function of the hyperparameters $\boldsymbol{\theta}$, whilst different acquisition functions $\alpha(\boldsymbol{\theta}; f)$ are proposed to balance between exploration and exploitation. This has been shown to give superior performance compared to traditional methods [Snoek et al., 2012] such as grid search or random search. However, BO suffers from the so called 'cold start' problem [Poloczek et al., 2016; Swersky et al., 2013], namely, initial observations of f at different hyperparameters are required to fit a GP model. Various methods [Swersky et al., 2013; Feurer et al., 2018; Springenberg et al., 2016; Poloczek et al., 2016] were proposed to address this issue by transferring knowledge from previously solved tasks, however, initial random evaluations of the models are still needed to consider the similarity across tasks. This might be prohibitive: evaluations of f can be computationally costly and our goal may be to select hyperparameters and deploy our model as soon as possible. We note that treating f as a black-box function, as is often the case in BO, is ignoring the highly structured nature of hyperparameter learning – it corresponds to training specific models on specific datasets. We make steps towards utilising such structure in order to borrow strength across different tasks and datasets.

Contribution We consider a scenario where a number of tasks have been previously solved and we propose a new BO algorithm, making use of the embeddings of the distribution of the training data [Blanchard et al., 2017; Muandet et al., 2017]. In particular, we propose a model that can jointly model all tasks at once, by considering an extended domain of inputs to model accuracy f, namely the distribution of the training data P_{XY}, sample size of the training data N and hyperparameters $\boldsymbol{\theta}$. Through utilising *all* seen evaluations from all tasks and meta-information, our methodology is able to learn a useful representation of the task that enables appropriate transfer of information to new tasks. As part of our contribution, we adapt our modelling approach to recent advances in scalable hyperparameter transfer learning [Perrone et al., 2018] and demonstrate that our proposed methodology can scale linearly in the number of function evaluations. Empirically, across a range of regression and classification tasks, our methodology performs favourably at initialisation and has a faster convergence compared to existing baselines – in some cases, the optimal accuracy is achieved in just a few evaluations.

3.2 Related work

The idea of transferring information from different tasks in the context of hyperparameter learning has been studied in various settings [Swersky et al., 2013; Feurer et al., 2018; Springenberg et al., 2016; Poloczek et al., 2016; Wistuba et al., 2018; Perrone et al., 2018]. Amongst this literature, one common feature is that the similarity across tasks is captured only through the evaluations of f. This implies that having sufficient evaluations from the task of interest is *necessary*, before we can transfer information. This is problematic, if model training is computationally expensive and our goal is to employ our model as quickly as possible. Further, the hyperparameter search for a machine learning model in general is not a black-box function, as we have additional information available: the dataset used in training. In our work, we aim to learn feature representation of training datasets in-order to yield good initial hyperparameter candidates without having seen any evaluations from our target task.

While such use of such dataset features, called *meta-features*, has been previously explored, current literature focuses on handcrafted meta-features[1]. These strategies are not optimal, as these meta-features can be be very similar, while having very different fs, and vice versa. In fact a study on OpenML [Vanschoren et al., 2013] meta-features have shown that the optimal set depends on the algorithm and data [Todorovski et al., 2000]. This suggests that the reliance on these features can have an adverse effect on exploration, and we give an example of this in Section 3.6. To avoid such shortcomings, given the same input space, our algorithm is able to *learn* meta-features directly from the data, avoiding such potential issues.

Although Kim et al. [2018] previously have also proposed to learn the meta-feature representations (for image data specifically), their proposed methodology requires the same set of hyperparameters to be evaluated for all previous tasks. This is clearly a *limitation* considering that different hyperparameter regions will be of interest for different tasks, and we would thus require excessive exploration of all those different regions under each task. To utilise meta-features, Kim et al. [2018] propose to warm-start Bayesian optimisation [Gomes et al.,

[1] A comprehensive survey on meta-learning and handcrafted meta-features can be found in [Hutter et al., 2019, Ch.2], [Feurer et al., 2015]

2012; Reif et al., 2012; Feurer et al., 2015] by initialising with the best hyperparameters from previous tasks. This also might be sub-optimal as we neglect non-optimal hyperparameters that can still provide valuable information for our new task, as we demonstrate in Section 3.6. Our work can be thought of to be similar in spirit to Klein et al. [2017], which considers an additional input to be the sample size N, but do not consider different tasks corresponding to different training data distributions.

3.3 Background

Our goal is to find $\boldsymbol{\theta}^*_{\text{target}} = \text{argmax}_{\boldsymbol{\theta}\in\Theta} f^{\text{target}}(\boldsymbol{\theta})$ where f^{target} is the target task objective we would like to optimise with respect to hyperparameters $\boldsymbol{\theta}$. In our setting, we assume that there are n (potentially) related source tasks $f^a, a = 1, \dots n$, and for each f^a, we assume that we have $\{\boldsymbol{\theta}^a_\ell, z^a_\ell\}_{\ell=1}^{s_a}$ from past runs, where z^a_ℓ denotes a noisy evaluation of $f^a(\boldsymbol{\theta}^a_\ell)$ and s_a denotes the number of evaluations of f^a from task a. Here, we focus on the case that $f^a(\boldsymbol{\theta})$ is some standardised accuracy (e.g. test set AUC) of a trained machine learning model with hyperparameters $\boldsymbol{\theta}$ and training data $D_a = \{\mathbf{x}^a_i, y^a_i\}_{i=1}^{N_a}$, where $\mathbf{x}^a_i \in \mathbb{R}^d$ are the covariates, y^a_i are the labels and N_a is the sample size of the training data. For a general framework, D_a is any input to f^a apart from $\boldsymbol{\theta}$ (can be unsupervised) – but following a typical supervised learning treatment, we assume it to be an i.i.d. sample from the joint distribution P_{XY}. For each task we now have:

$$(f^a, D_a = \{\mathbf{x}^a_i, y^a_i\}_{i=1}^{N_a}, \{\boldsymbol{\theta}^a_\ell, z^a_\ell\}_{\ell=1}^{s_a}), \quad a = 1, \dots n.$$

Our strategy now is to measure the similarity between datasets (as a representation of the task itself), in order to transfer information from previous tasks to help us quickly locate $\boldsymbol{\theta}^*_{\text{target}}$. In order to construct meaningful representations and measure distance between different tasks, we will make the assumption that $\mathbf{x}^a_i \in \mathcal{X}$ and $y^a_i \in \mathcal{Y}$ for all a, and that throughout the supervised learning model class is the same. While this setting might seem limiting, Feurer et al. [2018] and Poloczek et al. [2016] provides examples of many practical applications, including ride-sharing, customer analytic model, online inventory system and stock returns prediction. In all these cases, as new data becomes available, we might want to either re-train

our model (and find new optimal hyperparameters) or re-fit our parameters of the system to adapt to a specific distributional data input.

Intuitively, this assumption implies that the source of differences of $f^a(\boldsymbol{\theta})$ across a and $f^{\text{target}}(\boldsymbol{\theta})$ is in the data D_a and D_{target}. To model this, we will decompose the data D_a into the joint distribution P^a_{XY} of the training data ($D_a = \{\mathbf{x}^a_i, y^a_i\}_{i=1}^{N_a} \overset{i.i.d.}{\sim} P^a_{XY}$) and the sample size N_a for task a. Sample size[2] is important here as it is closely related to model complexity choice which is in turn closely related to hyperparameter choice [Klein et al., 2017]. While we have chosen to model D_a as P^a_{XY} and N_a, in practice through simple modifications of the methodology we propose, it is possible to model D_a as a set [Zaheer et al., 2017]. Under this setting, we will consider $f(\boldsymbol{\theta}, P_{XY}, N)$, where f is a function on hyperparameters $\boldsymbol{\theta}$, joint distribution P_{XY} and sample size N. For example, f could be the negative empirical risk, i.e.

$$f(\boldsymbol{\theta}, P_{XY}, N) = -\frac{1}{N}\sum_{i=1}^{N} L(h_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i),$$

where L is the loss function and $h_{\boldsymbol{\theta}}$ is the model's predictor. To recover f^a and f^{target}, we can evaluate at the corresponding P_{XY} and N, i.e. $f^a(\boldsymbol{\theta}) = f(\boldsymbol{\theta}, \mathcal{P}^a_{XY}, N_a)$, $f^{\text{target}}(\boldsymbol{\theta}) = f(\boldsymbol{\theta}, \mathcal{P}^{\text{target}}_{XY}, N_{\text{target}})$. In this form, we can see that similar to assuming that f varies smoothly as a function of $\boldsymbol{\theta}$ in standard BO, this model also assumes smoothness of f across P_{XY} as well as across N following Klein et al. [2017]. Here we can see that if two distributions and sample sizes are similar (with respect to a distance of their representations that we will learn), their corresponding values of f will also be similar. In this source and target task setup, this would suggest we can selectively utilise information from previous source datasets evaluations $\{\boldsymbol{\theta}^a_\ell, z^a_\ell\}_{\ell=1}^{s_a}$ to help us model f^{target}.

3.4 Methodology

3.4.1 Embedding of data distributions

To model P_{XY}, we will construct $\psi(D)$, a feature map on joint distributions for each task, estimated through its task's training data D. Here, we will follow similarly to Blanchard

[2]Following Klein et al. [2017], during implementation we re-scale N to $[0, 1]$, so that the task with the largest sample size has $N = 1$.

et al. [2017] which considers transfer learning, and make use of kernel mean embedding to compute feature maps of distributions (overview in Section 1.2.1.2 in Chapter 1). We begin by considering various feature maps of covariates and labels, denoting them by $\phi_x(\mathbf{x}) \in \mathbb{R}^q$, $\phi_y(y) \in \mathbb{R}^u$ and $\phi_{xy}([\mathbf{x}, y]) \in \mathbb{R}^c$, where $[\mathbf{x}, y]$ denotes the concatenation of covariates $\mathbf{x}$ and label y. Depending on the different scenarios, different quantities will be of interest.

Marginal Distribution P_X Modelling of the marginal distribution P_X is useful, as we might expect various tasks to differ in the distribution of $\mathbf{x}$ and hence in the hyperparameters $\boldsymbol{\theta}$, which, for example, may be related to the scales of covariates. We also might find that $\mathbf{x}$ is observed with different levels of noise across tasks. In this situation, it is natural to expect that those tasks with more noise would perform better under a simpler, more robust model (e.g. by increasing ℓ_2 regularisation in the objective function). To embed P_X, we can estimate the kernel mean embedding μ_{P_X} [Smola et al., 2007] with D by:

$$\psi(D) = \widehat{\boldsymbol{\mu}}_{P_X} = \frac{1}{N}\sum_{i=1}^{N} \phi_x(\mathbf{x}_i) \tag{3.1}$$

where $\psi(D) \in \mathbb{R}^q$ is an estimator of a representation of the marginal distribution P_X.

Conditional Distribution $P_{Y|X}$ Similar to P_X, we can also embed the conditional distribution $P_{Y|X}$. This is an important quantity, as across tasks, the form of the signal can shift. For example, we might have a latent variable W that controls the smoothness of a function, i.e. $P^a_{Y|X} = P_{Y|X,W=w_a}$. In a ridge regression setting, we will observe that those tasks (functions) that are more smooth would require a larger regularisation λ in order to perform better. For regression, to model the conditional distribution, we will use the kernel conditional mean operator $C_{Y|X}$ [Song et al., 2013] estimated with D by:

$$\widehat{\mathbf{C}}_{Y|X} \;=\; \boldsymbol{\Phi}_{\mathbf{y}}^\top(\boldsymbol{\Phi}_{\mathbf{x}}\boldsymbol{\Phi}_{\mathbf{x}}^\top + \lambda\mathbf{I})^{-1}\boldsymbol{\Phi}_{\mathbf{x}} = \lambda^{-1}\boldsymbol{\Phi}_{\mathbf{y}}^\top(\mathbf{I} - \boldsymbol{\Phi}_{\mathbf{x}}(\lambda\mathbf{I} + \boldsymbol{\Phi}_{\mathbf{x}}^\top\boldsymbol{\Phi}_{\mathbf{x}})^{-1}\boldsymbol{\Phi}_{\mathbf{x}}^\top)\boldsymbol{\Phi}_{\mathbf{x}} \tag{3.2}$$

where $\boldsymbol{\Phi}_{\mathbf{x}} = [\phi_x(\mathbf{x}_1), \ldots, \phi_x(\mathbf{x}_N)]^\top \in \mathbb{R}^{N\times q}$, $\boldsymbol{\Phi}_{\mathbf{y}} = [\phi_y(y_1), \ldots, \phi_y(y_N)]^\top \in \mathbb{R}^{N\times u}$ and λ is a regularisation parameter that we learn. It should be noted the second equality [Rasmussen & Williams, 2006] here allows us to avoid the $\mathcal{O}(N^3)$ arising from the inverse. This is important, as the number of samples N per task can be large. As $\widehat{\mathbf{C}}_{Y|X} \in \mathbb{R}^{u\times q}$, we will flatten it to obtain $\psi(D) \in \mathbb{R}^{qu}$ to obtain a representation of $P_{Y|X}$. In practice, as we rarely have prior insights into which quantity is useful for transferring hyperparameter information,

we will model both the marginal and conditional distributions together by concatenating the two feature maps above. The advantage of such an approach is that the learning algorithm does not have to itself decouple the overall representation of training dataset into information about marginal and conditional distributions which is likely to be informative.

Joint Distribution P_{XY} Taking an alternative and a more simplistic approach, it is also possible to model the joint distribution P_{XY} directly. One approach is to compute the kernel mean embedding, based on concatenated samples $[\mathbf{x}, y]$, considering the feature map ϕ_{xy}. Alternatively, we can also embed P_{XY} using the cross covariance operator C_{XY} [Gretton, 2015], estimated by D with:

$$\widehat{\mathbf{C}}_{XY} = \frac{1}{N}\sum_{i=1}^{N} \phi_x(\mathbf{x}_i) \otimes \phi_y(y_i) = \frac{1}{N}\Phi_{\mathbf{x}}^{\top}\Phi_{\mathbf{y}} \in \mathbb{R}^{q\times u}. \tag{3.3}$$

where $\otimes$ denotes the outer product and similar to $\widehat{\mathbf{C}}_{Y|X}$, we flatten it to obtain $\psi(D) \in \mathbb{R}^{qu}$. An important choice when modelling these quantities is the form of feature maps ϕ_x, ϕ_y and ϕ_{xy}, as these define the corresponding features of the data distribution we would like to capture. For example $\phi_x(\mathbf{x}) = \mathbf{x}$ and $\phi_x(\mathbf{x}) = \mathbf{x}\mathbf{x}^{\top}$ would be capturing the respective mean and second moment of the marginal distribution P_x. However, instead of defining a fixed feature map, here we will opt for a flexible representation, specifically in the form of neural networks (NN) for ϕ_x, ϕ_y and ϕ_{xy} (except ϕ_y for classification[3]), in a similar fashion to Wilson et al. [2016]. To provide a better intuition on this choice, suppose we have two tasks and that $P^1_{XY} \approx P^2_{XY}$ (with the same sample size N). This will imply that $f^1 \approx f^2$, and hence $\boldsymbol{\theta}^*_1 \approx \boldsymbol{\theta}^*_2$. However, the converse does not hold in general: $f^1 \approx f^2$ does *not* necessary imply $P^1_{XY} \approx P^2_{XY}$. For example, regularisation hyperparameters of a standard machine learning model are likely to be robust to rotations and orthogonal transformations of the covariates (leading to a different P_X). Hence, it is important to define a versatile model for $\psi(D)$, which can yield representations invariant to variations in the training data irrelevant for hyperparameter choice.

[3]For classification, we use $\widehat{\mathbf{C}}_{XY}$ and a one-hot encoding for ϕ_y implying a marginal embedding per class.

3.4.2 Modelling f

Given $\psi(D)$, we will now construct a model for $f(\boldsymbol{\theta}, P_{XY}, N)$, given observations of the form:

$$\{\{(\boldsymbol{\theta}_\ell^a, P_{XY}^a, N_a), z_\ell^a\}_{\ell=1}^{s_a}\}_{a=1}^n \tag{3.4}$$

along with any observations on the target. Note that we will interchangeably use the notation f to denote the model and the underlying function of interest. We will now focus on the algorithms distGP and distBLR, with additional details to be found in Appendix 3.8.1.

Gaussian Processes (distGP) We proceed similarly to standard BO [Snoek et al., 2012] using a GP (overview in Section 1.2.2.2 in Chapter 1) to model f and a normal likelihood (with variance σ^2 across all tasks[4]) for our observations z,

$$f \sim \text{GP}(m, k(\cdot,\cdot)) \qquad z|\gamma \sim \mathcal{N}(f(\gamma), \sigma^2) \tag{3.5}$$

where here m is the constant function, k is the corresponding covariance function on inputs $(\boldsymbol{\theta}, P_{XY}, N)$ with γ denoted a particular instance of the input. In order to fit a GP with inputs $(\boldsymbol{\theta}, P_{XY}, N)$, we use the following k:

$$k(\{\boldsymbol{\theta}_1, \mathcal{P}_{XY}^1, N_1\}, \{\boldsymbol{\theta}_2, \mathcal{P}_{XY}^2, N_2\}) = \nu k_{\boldsymbol{\theta}}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) k_P([\psi(D_1), N_1], [\psi(D_2), N_2]) \tag{3.6}$$

where ν is a constant, $k_{\boldsymbol{\theta}}$ and k_P is the standard Matérn-$3/2$ kernel (with separate bandwidths across the dimensions). For classification, we additionally concatenate the class size ratio per class, as this is not captured in $\psi(D_a)$. Utilising $\{\{(\boldsymbol{\theta}_\ell^a, P_{XY}^a, N_a), z_\ell^a\}_{\ell=1}^{s_a}\}_{a=1}^n$, we can optimise m, ν, σ^2 and any parameters in $\psi(D)$, $k_{\boldsymbol{\theta}}$ and k_P using the marginal likelihood of the GP (in an end-to-end fashion).

Bayesian Linear Regression (distBLR) While GP with its well-calibrated uncertainties have shown superior performance in BO [Snoek et al., 2012], it is well known that they suffer from $\mathcal{O}(S^3)$ computational complexity [Rasmussen & Williams, 2006], where $S = \sum_{a=1}^n s_a$ is the total number of observations. In this case, we might find that the total number of evaluations S across all tasks is too large for the GP inference to be tractable or that the computational burden of GPs outweighs the cost of computing f in the first place. To overcome

[4]For different noise levels across tasks, we can allow for different σ_a^2 per task a in distGP and distBLR.

this problem, we will follow Perrone et al. [2018] and use Bayesian linear regression (BLR), which scales linearly in the number of observations, with the model given by

$$z|\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\Upsilon}\boldsymbol{\beta}, \sigma^2\mathbf{I}) \qquad \boldsymbol{\beta} \sim \mathcal{N}(0, \alpha\mathbf{I}) \qquad \Psi_a = [\psi(D_a), N_a]$$

$$\boldsymbol{\Upsilon} = [\upsilon([\boldsymbol{\theta}_1^1, \Psi_1]), \ldots, \upsilon([\boldsymbol{\theta}_{s_1}^1, \Psi_1]), \ldots, \upsilon([\boldsymbol{\theta}_1^n, \Psi_n]), \ldots, \upsilon([\boldsymbol{\theta}_{s_n}^n, \Psi_n])]^\top \in \mathbb{R}^{S\times p} \tag{3.7}$$

where $\alpha > 0$ denotes the prior regularisation, and $[\cdot, \cdot]$ denotes concatenation. Here υ denotes a feature map on concatenated hyperparameters $\boldsymbol{\theta}$, data embedding $\psi(D)$ and sample size N. Following Perrone et al. [2018], we also employ a neural network for υ. While conceptually similar to Perrone et al. [2018] who fits a BLR per task, here we consider a single BLR fitted jointly on all tasks, highlighting differences across tasks using meta-information available. The advantage of our approach is that for a given new task, we are able to utilise directly all previous information and one-shot predict hyperparameters without seeing *any* evaluations from the target task. This is especially important when our goal might be to employ our system with only a few evaluations from our target task. In addition, a separate target task BLR is likely to be poorly fitted given only a few evaluations. Similar to the GP case, we can optimise α, σ^2 and any unknown parameters in $\psi(D), \upsilon([\boldsymbol{\theta}, \Psi])$ using the marginal likelihood of the BLR.

3.4.3 Hyperparameter learning

Having constructed a model for f and optimised any unknown parameters through the marginal likelihood, in order to construct a model for the f^{target}, we let $f^{\text{target}}(\boldsymbol{\theta}) = f(\boldsymbol{\theta}, \mathcal{P}_{XY}^{\text{target}}, N_{\text{target}})$. Now, to propose the next $\boldsymbol{\theta}^{\text{target}}$ to evaluate, we can simply proceed with Bayesian optimisation on f^{target}, i.e. maximise the corresponding acquisition function $\alpha(\boldsymbol{\theta}; f^{\text{target}})$. While we adopt standard BO techniques and acquisition functions here (an overview in Section 1.2.2.3 in Chapter 1), note that the generality of the developed framework allows it to be readily combined with many advances in the BO literature, e.g. Hernández-Lobato et al. [2014], Oh et al. [2018], McLeod et al. [2018], Snoek et al. [2012] and Wang et al. [2016].

Acquisition Functions For the form of the acquisition function $\alpha(\boldsymbol{\theta}; f^{\text{target}})$, we will use the popular expected improvement (EI) [Močkus, 1975]. However, for the first iteration, EI is not appropriate in our context, as these acquisition functions can favour $\boldsymbol{\theta}$ with high

uncertainty. Recalling that our goal is to quickly select 'good' hyperparameters $\boldsymbol{\theta}$ with few evaluations, for the *first* iteration we will maximise the lower confidence bound (LCB):

$$\alpha_{\text{LCB}}(\gamma; f_{\text{post}}) = \mu_{\text{post}}(\gamma) - \kappa\sigma_{\text{post}}(\gamma) \tag{3.8}$$

where $\kappa = 2.58$ denotes the level of exploration, as we would like to *exploit* the information from other tasks on our first iteration. It should be noted that this is not the upper confidence bound commonly used, as we want to *exploit* and obtain a good starting initialisation. While this approach works well for the GP case, for BLR, we will use the LCB restricted to the best hyperparameters from previous tasks, as BLR with a NN feature map does not extrapolate as well as GPs in the first iteration. For the exact forms of these acquisition functions, implementation and alternative warm-starting approaches, please refer to Appendix 3.8.2.

Optimisation We make use of ADAM [Kingma & Ba, 2015] to maximise the marginal likelihood until convergence. To ensure relative comparisons, we standardised each task's dataset features to have mean 0 and variance 1 (except for the unsupervised toy example), with regression labels normalised individually to be in $[0, 1]$. As the sample size per task N_a is likely to be large, instead of using the full set of samples N_a to compute $\psi(D_a)$, we will use a different random sub-sample of batch-size b for each iteration of optimisation. In practice, this parameter b is dependent on the number of tasks, and the evaluation cost of f. It should be noted that a smaller batch-size b would still provide an unbiased estimate of $\psi(D_a)$. At testing time, it is also possible to use a sub-sample of the dataset to avoid any computational costs arising from a large $\sum_{a=1}^{n} N_a$. When retraining, we will initialise from the previous set of parameters, hence few gradient steps are required before convergence occurs.

Extension to other data structures Throughout the chapter, we focused on examples with $\mathbf{x} \in \mathbb{R}^p$. However our formulation is more general, as we only require the corresponding feature maps to be defined on individual covariates and labels. For example, image data can be modelled by taking $\phi_x(\mathbf{x})$ to be a representation given by a convolutional neural network (CNN)[5], while for text data, we might construct features using Word2vec [Mikolov et al., 2013], and then retrain these representations for hyperparameter learning setting. More broadly, we can initialise $\psi(D)$ to any meaningful representation of the data, believed to

[5]This is similar to Law et al. [2018b] who embeds distribution of images using a pre-trained CNN for distribution regression.

be useful to the selection of $\boldsymbol{\theta}^*_{\text{target}}$. Of course, we can also choose $\psi(D)$ simply as a selection of handcrafted meta-features [Hutter et al., 2019, Ch. 2], in which case our methodology would use these representations to measure similarity between tasks, while performing feature selection [Todorovski et al., 2000]. In practice, learnt feature maps via kernel mean embeddings can be used in conjunction with handcrafted meta-features, letting the data speak for itself. In Section 3.5.1, we provide a selection of 13 handcrafted meta-features that we employ as a baseline in experiments.

3.5 Alternative approaches

Here we will discuss the various baselines that we employ to demonstrate the effectiveness of our algorithm.

3.5.1 manualBO

Instead of constructing $\psi(D)$, as described in Section 3.4.1, we can take $\psi(D)$ to be a selection of handcrafted meta-features. To ensure fair relative comparisons, the features $\{\mathbf{x}_i^a\}_{i=1}^{N_a}$ are standardised to have mean 0 and variance 1 and the meta-features are normalised to be in $[0, 1]$ across all tasks [Bardenet et al., 2013]. We do not include sample size N_a, as these are already encoded separately.

General meta-features

- Skewness, kurtosis [Michie et al., 1994]: these are calculated on each feature of the dataset D_a, before the minimum, maximum, mean and standard deviation of the computed quantities is extracted across the features.
- Correlation, covariance [Michie et al., 1994]: these are calculated on every pair of features of D_a, before the minimum, maximum, mean and standard deviation of the computed quantities is extracted across each pair of features.
- PCA skewness, kurtosis [Feurer et al., 2014]: principal component analysis (PCA) is performed on D_a, and D_a is projected onto the first principal component. The corresponding skewness and kurtosis is computed.

- Intrinsic dimensionality [Bardenet et al., 2013]: number of principal components to explain 95% of variance.

Classification specific meta-features

- Class ratios, entropy [Michie et al., 1994]: empirical class distribution and its corresponding entropy.
- Classification landmarkers [Pfahringer et al., 2000]: 1-nearest-neighbour classifier, linear discriminant analysis, naive Bayes and decision tree classifier.

Regression specific meta-features

- Mean, standard deviation, skewness, kurtosis of the labels $\{y_i^a\}_{i=1}^{N_a}$ [Michie et al., 1994].
- Regression landmarkers [Pfahringer et al., 2000]: 1-nearest-neighbour regressor, linear regression and decision tree regressor.

The landmarkers are scalable algorithms that are cheap to run, and provide us various characteristic of the machine learning task. The corresponding meta-feature from these landmarkers is the accuracy on an independent set of data (a train-test split is done on B_a, the training data). In experiments, we use the default settings in *sklearn* [Pedregosa et al., 2011] for these algorithms. For additional details on their formulation and rationale, please refer to [Hutter et al., 2019, Ch.2].

3.5.2 multiBO

Instead of using meta-features, we may wish to simply encode the task index, and learn task similarities based on only $\{\{\boldsymbol{\theta}_\ell^a, z_i^a\}_{\ell=1}^{s_a}\}_{a=1}^n$. Here, we do not encode any sample size or class ratio information and it is noted that initial evaluations from the target task is required.

multiGP For the GP case, we will follow Swersky et al. [2013], who considers a multi-task GP for Bayesian optimisation. Instead of using the kernel k_P on meta-features, we will now replace it by a kernel on tasks k_t. Given the $n+1$ total number of tasks (including the target task), the task similarity matrix is given by $\mathbf{S}_t = \mathbf{L}_t\mathbf{L}_t^T \in \mathbb{R}^{n+1\times n+1}$, where $\mathbf{L}_t$ is a learnt cholesky factor. Expanding $\mathbf{S}_t$ into the appropriate sized kernel $\mathbf{K}_t \in \mathbb{R}^{S\times S}$ (as we

have repeated observations from the same task), using the marginal likelihood, we can learn the lower triangular elements of $\mathbf{L}_t$. Similar to Swersky et al. [2013], we assume positive correlation amongst tasks and restrict positivity in the elements of the cholesky factor.

multiBLR For the BLR case, we will follow Perrone et al. [2018] and consider a one-hot encoding for $\psi(D_a)$. This representation essentially identifies a separate encoding for every task, and similarity between tasks (and hyperparameters) is captured through the transformation υ (without sample size N_a), which we learn using the marginal likelihood.

3.5.3 initBO

For this baseline, we will employ the handcrafted meta-features as described in Section 3.5.1 to warm-start Bayesian optimisation, using a GP or BLR. In particular, we first define the number of evaluations m per task and the number of tasks M we wish to warm-start with (i.e. Mm number of warm-start hyperparameters). To define a similarity function, for a fair comparison with existing literature, we will use the L_2 norm [Feurer et al., 2015] between the datasets' meta-features:

$$k(D_k, D_j) = -|| \, [\psi(D_k), N_k] - [\psi(D_j), N_j] \, ||_2$$

where here k is a similarity function, and $\psi(D_a)$ is the handcrafted meta-features representation for task a. It should also be noted that as meta-features are individually normalised to be in $[0, 1]$, no particular meta-feature is emphasised in this distance measure. To obtain the warm-start $\boldsymbol{\theta}$s, we compute $k(D_{\text{target}}, D_a)$ for all $a = 1, \ldots, n$ and extract the M tasks with the highest similarity. Given these M tasks, we extract the m best performing hyperparameters from each of these task to obtain Mm warm-start hyperparameters. These hyperparameters will then be used for warm-starting the standard GP or BLR Bayesian optimisation (instead of random evaluations).

3.6 Experiments

We will denote our methodology distBO, with BO being a placeholder for GP and BLR versions. For ϕ_x and ϕ_y we will use a single hidden layer NN with $\tanh$ activation (with 20

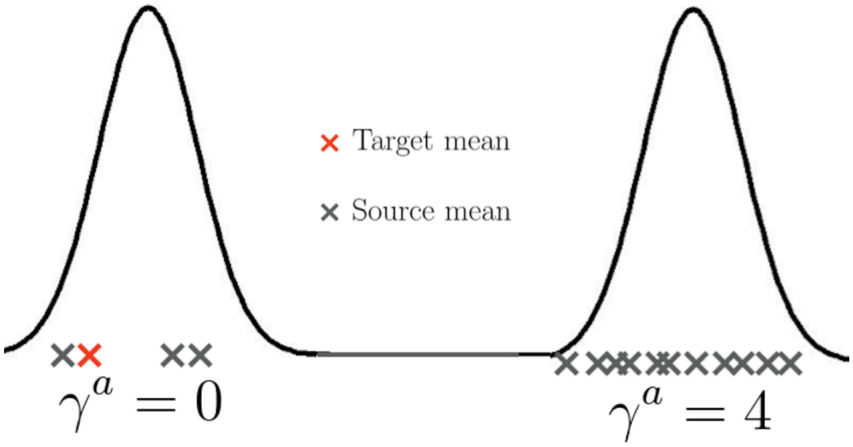

Figure 3.1: Illustration of unsupervised toy example.

hidden and 10 output units), except for classification tasks, where we use a one-hot encoding for ϕ_y. For clarity purposes, we will focus on the approach where we separately embed the marginal and conditional distributions, before concatenation. Additional results for embedding the joint distribution can be found in Appendix 3.8.3.1. For BLR, we will follow Perrone et al. [2018] and take feature map υ to be a NN with three 50-unit layers and $\tanh$ activation.

For baselines, we will consider: 1) manualBO with $\psi(D)$ as the selection of 13 handcrafted meta-features; 2) multiBO, i.e. multiGP [Swersky et al., 2013] and multiBLR [Perrone et al., 2018] where no meta-information is used, i.e. task is simply encoded by its index (they are initialised with 1 random iteration); 3) initBO [Feurer et al., 2015] with plain Bayesian optimisation, but warm-started with the top 3 hyperparameters, from the three most similar source tasks, computing the similarity with the L_2 distance on handcrafted meta-features; 4) noneBO denoting the plain Bayesian optimisation [Snoek et al., 2012], with no previous task information; 5) RS denoting the random search. In all cases, both GP and BLR versions are considered.

We use *TensorFlow* [Abadi et al., 2016] for implementation, repeating each experiment 30 times, either through re-sampling or re-splitting the train/test partition. For testing, we use the same number of samples N_a for toy data, while using a 60-40 train-test split for real data. We take the embedding batch-size[6] $b = 1000$, and learning rate for ADAM to be 0.005.

[6]Training time is less than 2 minutes on a standard 2.60GHz single-core CPU in all experiments.

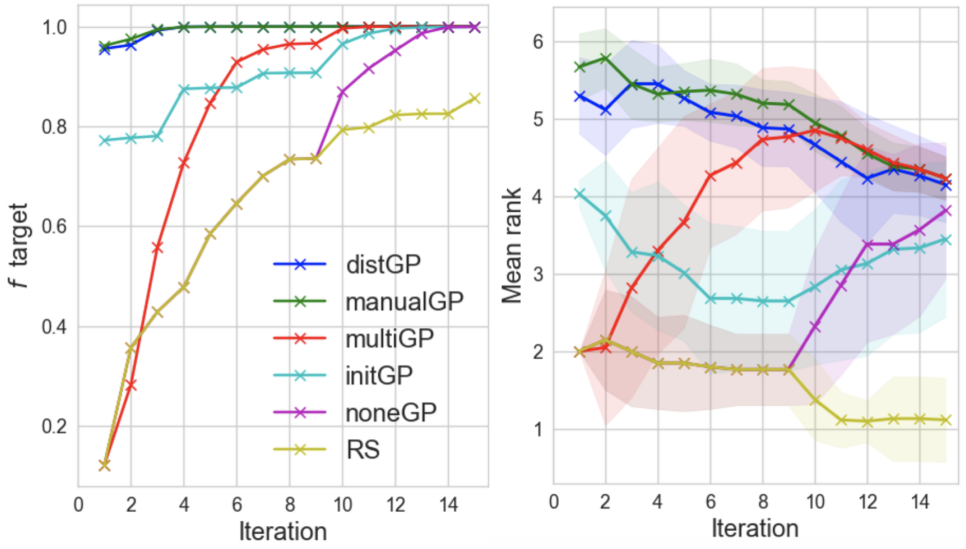

Figure 3.2: Unsupervised toy task with 15 iterations (including any initialisation). Each evaluation here is averaged over 30 runs. **Left:** *Maximum observed* f^{target}. **Right:** Mean rank (with respect to each run) of the different methodologies, with ± 1 sample standard deviation.

To obtain $\{\boldsymbol{\theta}_\ell^a, z_\ell^a\}_{\ell=1}^{s_a}$ for source task a, we use noneGP to simulate a realistic scenario. Additional details on experiments can be found in Appendix 3.8.3, with additional real life (*Parkinson's dataset*) experiments to be found in Appendix 3.8.3.4.

3.6.1 Toy example

To understand the various characteristics of the different methodologies, we first consider an unsupervised toy 1-dimensional example, where the dataset D_a follows the generative process for some fixed γ^a: $\mu^a \sim \mathcal{N}(\gamma^a, 1)$; $\{x_i^a\}_{i=1}^{N_a} | \mu^a \overset{i.i.d.}{\sim} \mathcal{N}(\mu^a, 1)$. We can think of μ^a as the (unobserved) relevant property varying across tasks, and the unlabelled dataset as $D_a = \{x_i^a\}_{i=1}^{N_a}$. Here, we will consider the objective f given by:

$$f(\theta; D_a) = \exp\left(-\frac{(\theta - \frac{1}{N_a}\sum_{i=1}^{N_a} x_i^a)^2}{2}\right), \tag{3.9}$$

where $\theta \in [-8, 8]$ plays the role of a 'hyperparameter' that we would like to select. Here, the optimal choice for task a is $\theta = \frac{1}{N_a}\sum_{i=1}^{N_a} x_i^a$ and hence it is varying together with the underlying mean μ^a of the sampling distribution. An illustration of this experiment can be found in Figure 3.1.

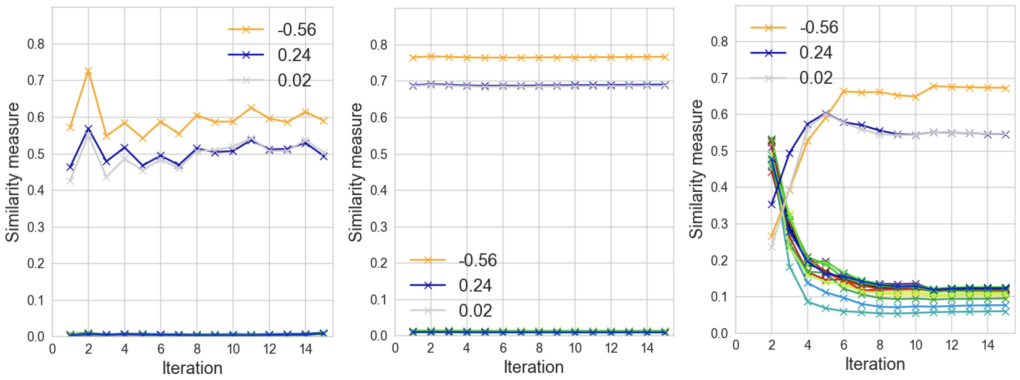

Figure 3.3: Mean of the similarity measure $k_P(\psi(D_a), \psi(D_{\text{target}}))$ over 30 runs versus number of iterations for the unsupervised toy task. For clarity purposes, the legend *only* shows the μ^a for the 3 source tasks that are similar to the target task with $\mu^a = -0.25$. It is noted the rest of the source task have $\mu^a \approx 4$. **Left:** distGP **Middle:** manualGP **Right:** multiGP

We now perform an experiment with $n = 15$, and $N_a = 500$, for all a, and generate 3 source tasks with $\gamma^a = 0$, and 12 source task with $\gamma^a = 4$. In addition, we generate an additional target dataset with $\gamma^{\text{target}} = 0$ and let the number of source evaluations per task be $s_a = 30$. The results can be found in Figure 3.2. Here, we observe that distBO has correctly learnt to utilise the appropriate source tasks, and is able to few-shot the optimum. This is also evident on the left of Figure 3.3, which shows the similarity measure $k_P(\psi(D_a), \psi(D_{\text{target}})) \in [0, 1]$ for distGP. The feature representation has correctly learnt to place high similarity on the three source datasets sharing the same γ^a and hence having similar values of μ^a, while placing low similarity on the other source datasets.

As expected, manualBO also few-shots the optimum here since the mean meta-feature which directly reveals the optimal hyperparameter was explicitly encoded in the hand-crafted ones. initBO starts reasonably well, but converges slowly, since the optimal hyperparameters even in the similar source tasks are not the same as that of the target task. It is also notable that multiBO is unable to few-shot the optimum, as it does not make use of any meta-information, hence needing initialisation from the target task to even begin learning the similarity across tasks. This is especially highlighted in Figure 3.3, which shows an incorrect similarity in the first few iterations. Significance is shown in the mean rank graph on the right of Figure 3.2.

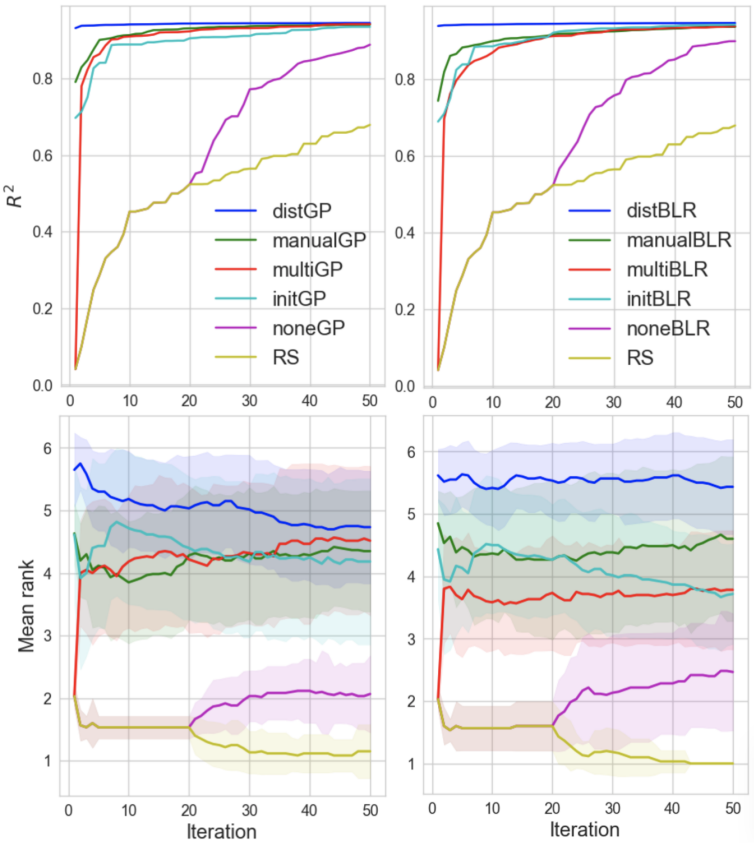

Figure 3.4: Manual meta-features counterexample with 50 iterations (including any initialisation) with GP (left) and BLR (right). Each evaluation here is averaged over 30 runs. **Top row:** *Maximum observed* R^2. **Bottom row:** Mean rank (with respect to each run) of the different methodologies, with ± 1 sample standard deviation.

3.6.2 Regression: Handcrafted meta-features fail

We now demonstrate an example in which using handcrafted meta-features does not capture any information about the optimal hyperparameters of the target task. Consider the following process for dataset a with $\mathbf{x}_i^a \in \mathbb{R}^6$ and $y_i^a \in \mathbb{R}$, given by:

$$\begin{aligned} [\mathbf{x}_i^a]_k &\overset{i.i.d.}{\sim} \mathcal{N}(0, 2^2), \quad k = 1, \ldots, 6, \\ [\mathbf{x}_i^a]_{a+2} &= \text{sign}([\mathbf{x}_i^a]_1 [\mathbf{x}_i^a]_2) \, |[\mathbf{x}_i^a]_{a+2}| \,, \\ y_i^a &= \log\left(1 + \left(\prod_{k \in \{1,2,a+2\}} [\mathbf{x}_i^a]_k\right)^3\right) + \epsilon_i^a. \end{aligned} \tag{3.10}$$

where $\epsilon_i^a \overset{iid}{\sim} \mathcal{N}(0, 0.5^2)$, with index a, i, k denoting task, sample and dimension, respectively: $a = 1, \ldots, 4$ and $i = 1, \ldots, N_a$ with sample size $N_a = 5000$. Thus across $n = 4$ source tasks, we have constructed regression problems, where the dimensions which are relevant

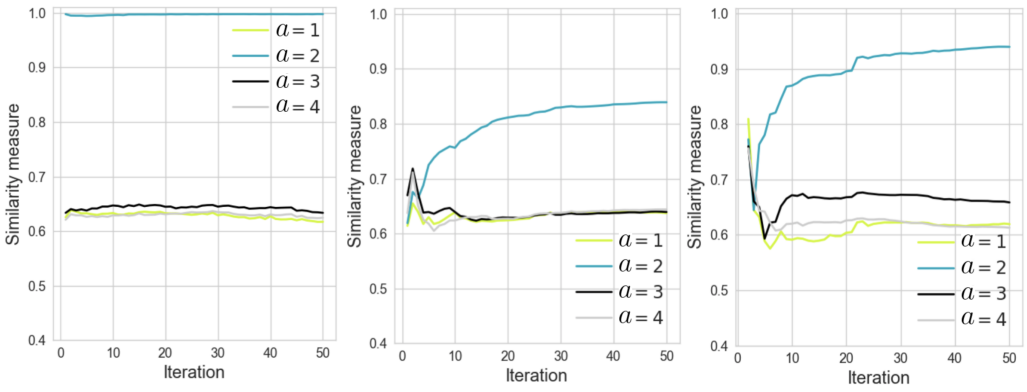

Figure 3.5: Mean of the similarity measure $k_P(\psi(D_a), \psi(D_{\text{target}}))$ over 30 runs versus number of iterations for the manual meta-features counterexample. The target task has the same generative process as $a = 2$. **Left:** distGP **Middle:** manualGP **Right:** multiGP

(namely 1, 2 and $a + 2$) are varying. Note that (3.10) introduces a three-variable interaction in the relevant dimensions, but that all dimensions remain pairwise independent and identically distributed, with $[\mathbf{x}_i^a]_{a+2} \sim \mathcal{N}(0, 2^2)$ even after alteration. Thus, while these tasks are inherently different, this difference is invisible by considering marginal distribution of covariates and their pairwise relationships such as covariances. Similarly, any PCA meta-features will remain the same, as variances remains the same in all directions. For regression landmarkers and labels, as these are not perturbed by permutation of the features of the dataset, the regression specific meta-features also remains the same. This implies that the handcrafted meta-features for manualBO which only consider statistics which process one or two features at a time or landmarkers [Pfahringer et al., 2000] have corresponding $\psi(D_a)$ that are *invariant* to tasks up to sampling variations[7].

We now generate an additional target dataset, using the same generative process as $a = 2$, and let f be the coefficient of determinant (R^2) on the test set resulting from an automatic relevance determination (ARD) kernel ridge regression with hyperparameters α and $\sigma_1, \ldots, \sigma_6$. Here α denotes the regularisation parameter, while σ_k denotes the kernel bandwidth for dimension k. Setting $s_a = 125$, the results can be found in Figure 3.4. It is clear that while distBO is able to learn a high similarity to the correct source task (as shown in Figure 3.5), and one-shot the optimum, this is not the case for any of the other baselines. In fact, as manualBO's meta-features do not include any useful meta-information, they essentially

[7]Sampling variations implies that the computed representation $\psi(D_a)$ still differs slightly amongst all the tasks, hence the specific task can still be recognised without the encoding of task index for manualBO.

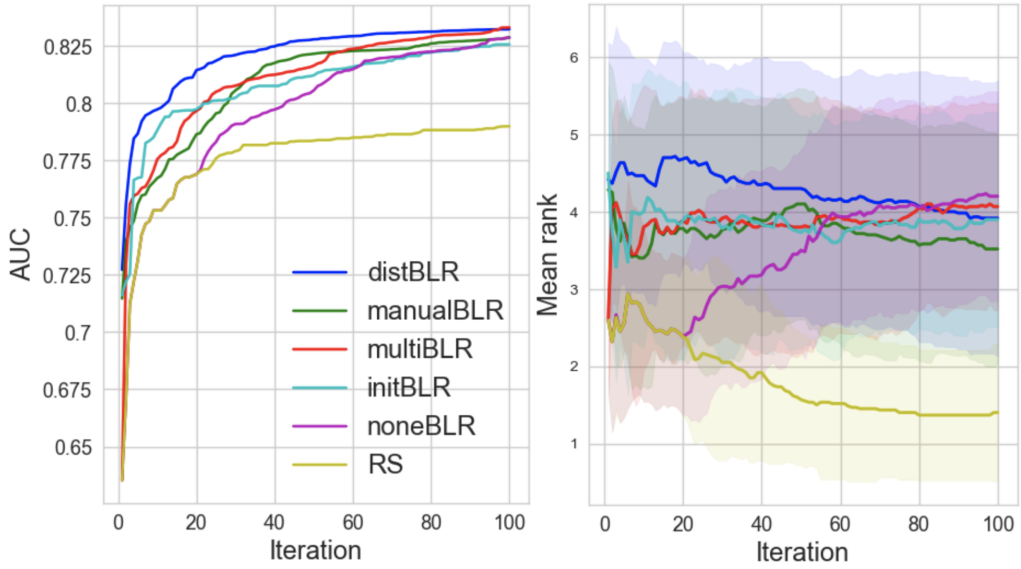

Figure 3.6: Classification task experiment A with 100 iterations (including any initialisation). Here, the target task is similar to one of the source task. Each evaluation here is averaged over 30 runs. **Left:** *Maximum observed* AUC. **Right:** Mean rank (with respect to each run) of the different methodologies, with ± 1 sample standard deviation.

encode the task index, and hence perform similarly to multiBO. Further, we observe that initBO has slow convergence after warm-starting. This is not surprising as initBO has to 're-explore' the hyperparameter space as it only uses a subset of previous evaluations. This highlights the importance of using all evaluations from all source tasks, even if they are sub-optimal. In the bottom row of Figure 3.4, we also show significance using a mean rank graph and that the BLR methods performs similarly to their GP counterparts.

3.6.3 Classification: Similar and not similar source tasks

We now demonstrate a classification example, where we contrast the case where some of the source tasks is similar to the target task against the case where no such source task exists to illustrate that encoding meta-information need not always be beneficial. Here, we let the number of source tasks $n = 10$, $N_a = 5000$ and f to be the AUC on the test set for ARD kernel logistic regression, with hyperparameters C and $\sigma_1, \ldots, \sigma_6$. Similar to before, C denotes regularisation and σ_k denotes the kernel bandwidth for dimension k. To generate D_a, we take $\mathbf{x}_i^a \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_6)$, and obtain y_i^a conditionally on $\mathbf{x}_i^a$ by sampling from a kernel logistic regression model (ARD kernel with Random Fourier features [Rahimi & Recht,

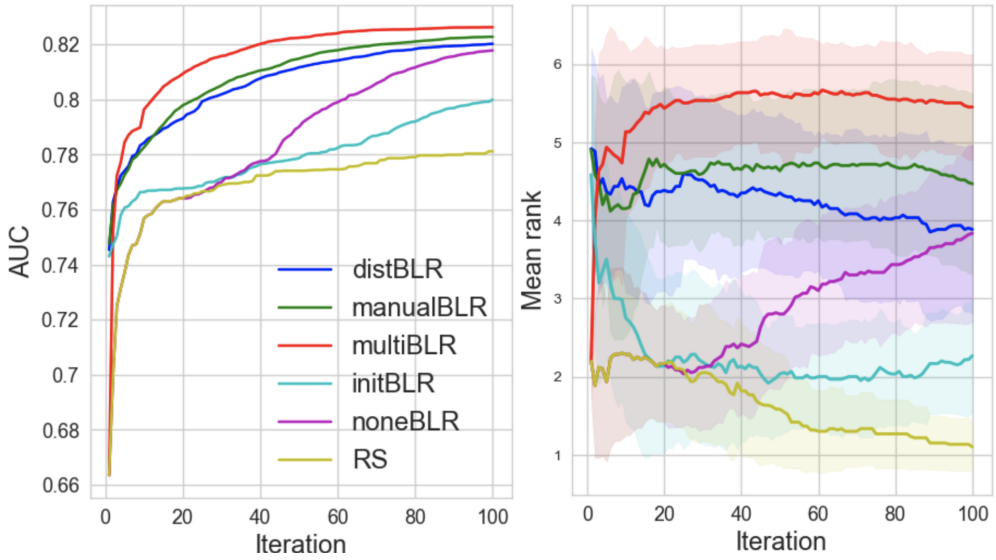

Figure 3.7: Classification task experiment B with 100 iterations (including any initialisation). Here the target task is *different* to all the source task. Each evaluation here is averaged over 30 runs. **Left:** *Maximum observed* AUC. **Right:** Mean rank (with respect to each run) of the different methodologies, with ± 1 sample standard deviation.

2007] approximation) where each task has different true bandwidth parameters (the generation process is discussed in Appendix 3.8.3.3). For the source tasks, we will randomly select $\tilde{\sigma}_k^a \in \{0.5, 1.0, 2.0, 4.0, 8.0, 16.0\}$ with replacement across all k, so that different dimensions are of different relative importance across different tasks. For experiment A, we will select its underlying bandwidths to be the same as one of that in the source task. For experiment B, to ensure that our target task has different optimal hyperparameters to the source tasks, we will let $\tilde{\sigma}_k^a = 1.5$ for all k.

Note that all tasks have the same marginal distribution of covariates and that there is a high variation in conditional distributions: they differ not only in terms of kernel bandwidths but also in terms of coefficients in their respective regression functions. To generate a task dataset, we use the same process, and run 2 experiments: (A) use the same set of bandwidths as one of the source tasks but a different regression function, and (B) use a set of bandwidths unseen in any of the source tasks (and a different regression function). We take $s_a = 150$ and since the total number of evaluations is $S = 1500$, we focus our attention on BLR, which have $\mathcal{O}(S)$ linear computational complexity. The results for the two experiments are shown in Figure 3.6 and 3.7.

We see that distBLR leverages the presence of a similar task among the sources and learns

a representation of the dataset which helps guide hyperparameter selection to the optimum faster than other methods. We note that manualBLR converges much slower, given that the optimal hyperparameters depend on the data in a complex way which is difficult to extract from handcrafted meta-features. We also note that initBLR performs poorly despite the presence of a source task with the *same true bandwidths*: often, the meta-features are not powerful enough to recognise which task is the most similar in order to initialise appropriately.

On the other hand, in the case B, no similar source exists implying that the joint BLR model in distBLR needs to extrapolate to the far away region in the space of joint distributions of the training data. As expected, meta-information in this example is not as helpful as in the case A and the method that ignores it, multiBLR, in fact performs best. However, albeit worse performing, note that distBLR and manualBLR were still able to revert to the behaviour akin to multiBLR and achieve a faster convergence compared to their non-transfer counterparts and initBLR which essentially has to re-explore the hyperparameter space from scratch.

3.6.4 Classification: Protein dataset

The Protein dataset consists of 7 proteins extracted from Gaulton et al. [2016]: ADAM17, AKT1, BRAF, COX1, FXA, GR, VEGFR2. Each dataset contains $1037 - 4434$ molecules (data-points N_a), where each molecule has binary features $\mathbf{x}_i^a \in \mathbb{R}^{166}$ computed using a chemical fingerprint (MACCs Keys[8]). The label per molecule is whether the molecule can bind to the protein target $\in \{0, 1\}$. In this experiment, we can treat each protein as a separate classification task.

We consider two classification methods: Jaccard kernel C-SVM [Bouchard et al., 2013; Ralaivola et al., 2005] (commonly used for binary data, with hyperparameter C), and random forest (with hyperparameters n_trees, max_depth, $min\_samples\_split$, $min\_samples\_leaf$), with the corresponding objective f for each given by accuracy rate on the test set. In this experiment, we will designate each protein as the target task, while using the other $n = 6$ proteins as source tasks. In particular, we will take $s_a = 20$ and hence $S = 120$. The results obtained by averaging over different proteins as the target task (20 runs per task) are shown

[8] http://rdkit.org/docs/source/rdkit.Chem.MACCSkeys.html

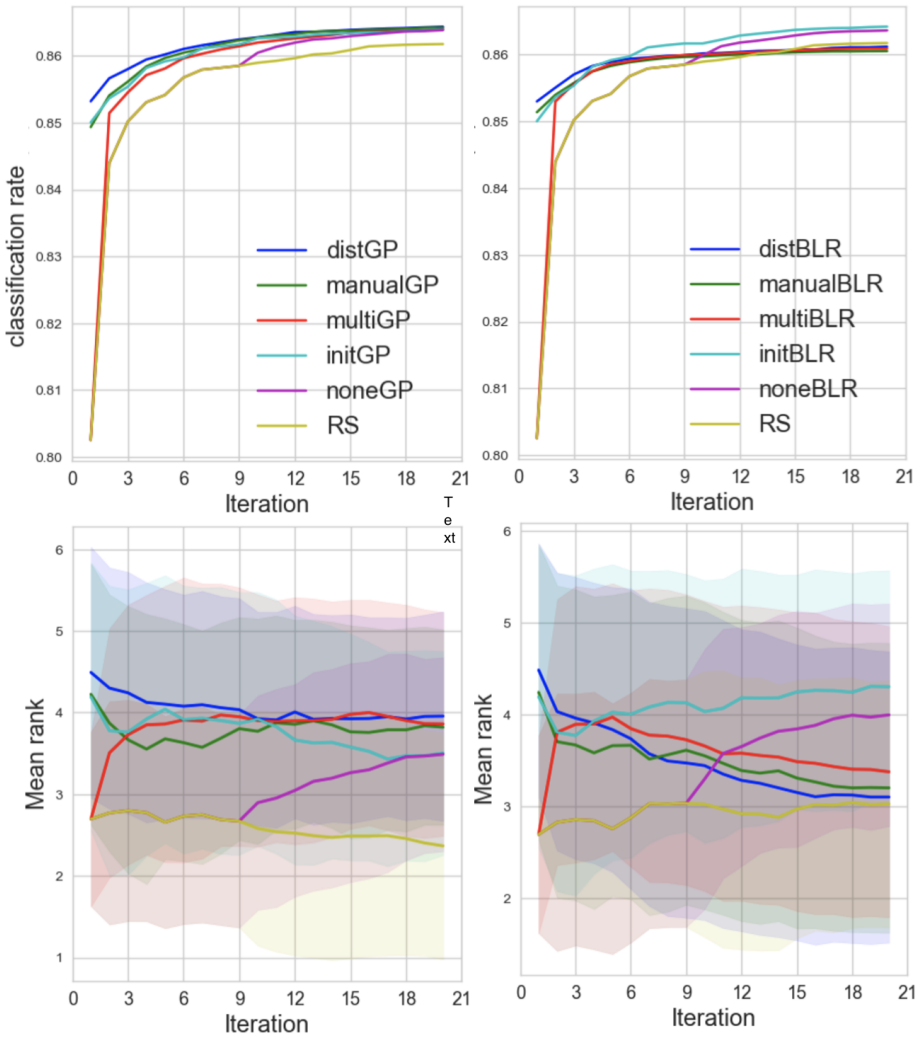

Figure 3.8: Protein dataset with Jaccard kernel C-SVM. Each evaluation here is averaged over 140 runs, with each of the 7 protein set as the target task (20 runs each). GP methods are displayed on the left, while BLR methods are displayed on the right. **Top row:** *Maximum observed* classification accuracy (%). **Bottom row:** Mean rank (with respect to each run) of the different methodologies, with ± 1 sample standard deviation.

in Figure 3.8 and 3.9. On this dataset, we observe that distGP and distBLR outperforms its counterpart baselines and few-shots the optimum for both algorithms. In addition, we can see a slower convergence for the multiGP and initGP, demonstrating the usefulness of meta information in this context.

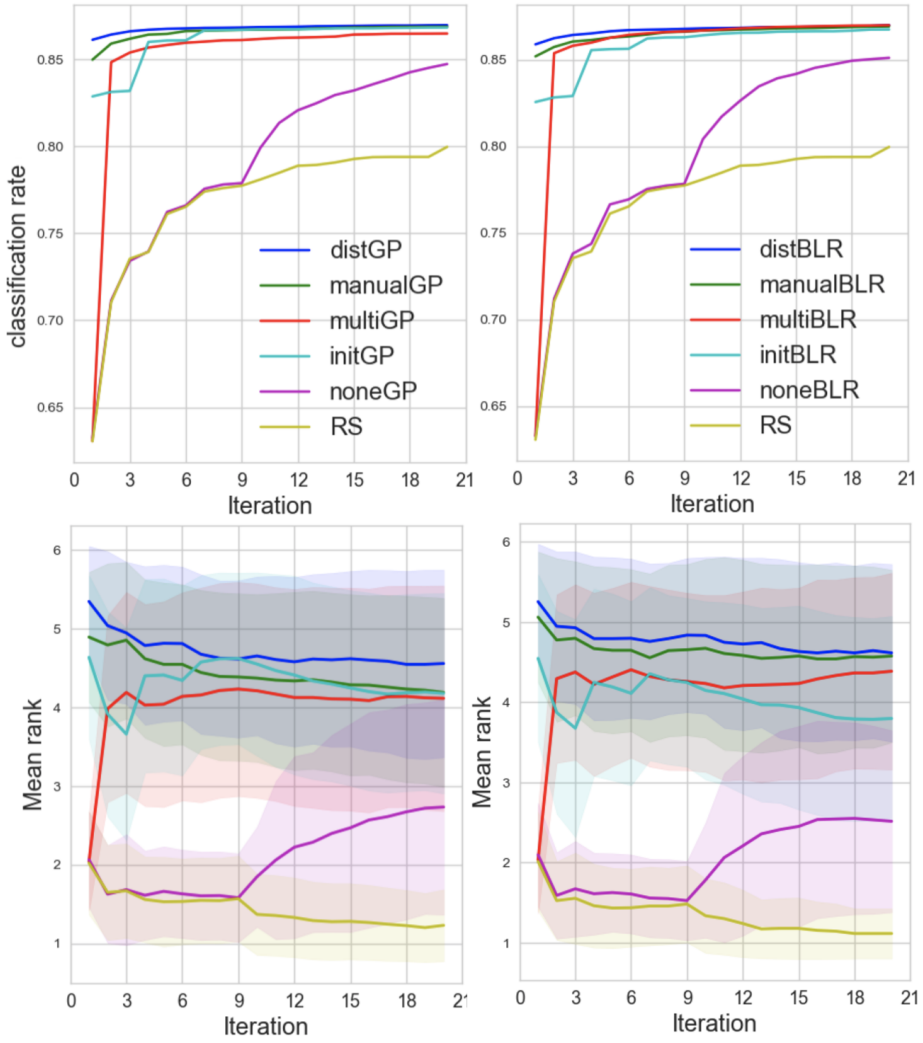

Figure 3.9: Protein dataset with random forest. Each evaluation here is averaged over 140 runs, with each of the 7 protein set as the target task (20 runs each). GP methods are displayed on the left, while BLR methods are displayed on the right. **Top row:** *Maximum observed* classification accuracy (%). **Bottom row:** Mean rank (with respect to each run) of the different methodologies, with ± 1 sample standard deviation.

3.7 Conclusion

We demonstrated that it is possible to borrow strength between multiple hyperparameter learning tasks by making use of the similarity between training datasets used in those tasks. This helped us to develop a method which finds a favourable setting of hyperparameters in only a few evaluations of the target objective. We argue that the model performance should not be treated as a black box function as it corresponds to specific known models and specific datasets and that its careful consideration as a function of all its inputs, and not just of its hyperparameters, can lead to useful algorithms.

3.8 Chapter appendix

3.8.1 Additional details for methodology

Gaussian process (distGP)

For distGP, we have the following model:

$$\begin{aligned} f &\sim \text{GP}(m(\cdot), k(\cdot,\cdot)) \\ z|\boldsymbol{\gamma} &\overset{i.i.d.}{\sim} \mathcal{N}(f(\boldsymbol{\gamma}), \sigma^2) \end{aligned}$$

where here $m(\cdot)$ is taken to be a constant function and $k(\cdot,\cdot)$ is the corresponding covariance function. The log marginal likelihood with observations $\Gamma = \{\{(\boldsymbol{\theta}_i^a, P_{XY}^a, N_a), z_\ell^a\}_{\ell=1}^{s_a}\}_{a=1}^n$, following standard GP literature [Rasmussen & Williams, 2006] is given by:

$$\log(p(\mathbf{z}|\Gamma)) = -\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^\top(\mathbf{K}+\sigma^2\mathbf{I})^{-1}(\mathbf{z}-\boldsymbol{\mu}) - \frac{1}{2}\log|\mathbf{K}+\sigma^2\mathbf{I}| - \frac{S}{2}\log(2\pi)$$

where $\mathbf{z} = [z_1^1, \dots z_{s_n}^n]^\top$, $S = \sum_{a=1}^n s_a$ and $\mathbf{K}$ is the kernel matrix, with $\mathbf{K}_{uv} = k(\boldsymbol{\gamma}_u, \boldsymbol{\gamma}_v)$. Here $\boldsymbol{\gamma}_u, \boldsymbol{\gamma}_v$ denotes elements of $\{\{\boldsymbol{\theta}_\ell^a, P_{XY}^a, N_a\}_{\ell=1}^{s_a}\}_{a=1}^n$. In particular, for a new observation $\boldsymbol{\gamma}^*$, the predictive posterior distribution $f_{\text{post}}(\boldsymbol{\gamma}^*) \sim \mathcal{N}(\mu_{\text{post}}(\boldsymbol{\gamma}^*), \sigma^2_{\text{post}}(\boldsymbol{\gamma}^*))$, where:

$$\begin{aligned} \mu_{\text{post}}(\boldsymbol{\gamma}^*) &= \boldsymbol{\mu} + \mathbf{K}_{\gamma^*\Gamma}(\mathbf{K}+\sigma^2\mathbf{I})^{-1}(\mathbf{z}-\boldsymbol{\mu}) \\ \sigma^2_{\text{post}}(\boldsymbol{\gamma}^*) &= \mathbf{K}_{\boldsymbol{\gamma}^*\boldsymbol{\gamma}^*} - \mathbf{K}_{\boldsymbol{\gamma}^*\Gamma}(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{K}_{\boldsymbol{\gamma}^*\Gamma}^\top \end{aligned}$$

where here $\mathbf{K}_{\boldsymbol{\gamma}^*\boldsymbol{\gamma}^*} = k(\boldsymbol{\gamma}^*, \boldsymbol{\gamma}^*)$ and $\mathbf{K}_{\boldsymbol{\gamma}^*\Gamma} = [k(\boldsymbol{\gamma}^*, \boldsymbol{\gamma}_1), \dots, k(\boldsymbol{\gamma}^*, \boldsymbol{\gamma}_S)]$.

Bayesian Linear Regression (distBLR)

$$z|\boldsymbol{\beta} \overset{i.i.d.}{\sim} \mathcal{N}(\boldsymbol{\Upsilon}\boldsymbol{\beta}, \sigma^2\mathbf{I}) \qquad \boldsymbol{\beta} \sim \mathcal{N}(0, \alpha\mathbf{I})$$

where $\boldsymbol{\Upsilon} = [\upsilon([\boldsymbol{\theta}_1^1, \psi(D_1), N_1]), \dots, \upsilon([\boldsymbol{\theta}_{s_n}^n, \psi(D_n), N_n])]^\top \in \mathbb{R}^{S\times p}$ and $\alpha > 0$ denotes the prior regularisation. Here $\boldsymbol{\Upsilon}$ denotes a matrix of feature maps of dimension p on concatenated hyperparameters $\boldsymbol{\theta}$, data embedding $\psi(D)$ and sample size N. Following Perrone et al. [2018], defining $\mathbf{K}_{\text{dim}} = \mathbf{I}_p + \frac{\alpha}{\sigma^2}\boldsymbol{\Upsilon}^\top\boldsymbol{\Upsilon}$, and $\mathbf{L}$ as the cholesky factor of $\mathbf{K}_{\text{dim}}$, i.e. $\mathbf{K}_{\text{dim}} = \mathbf{L}\mathbf{L}^\top$, the log marginal likelihood (up to additive constants) with observations $\Gamma = \{\{(\boldsymbol{\theta}_\ell^a, P_{XY}^a, N_a), z_\ell^a\}_{\ell=1}^{s_a}\}_{a=1}^n$ is given by:

$$\log(p(\mathbf{z}|\Gamma)) = \frac{1}{2\sigma^2}\left(\frac{\alpha}{\sigma^2}||\mathbf{e}||^2 - ||\mathbf{z}||^2\right) - \sum_{k=1}^p \log(\mathbf{L}_{kk}) - \frac{S}{2}\log(\sigma^2)$$

where $\mathbf{e} = \mathbf{L}^{-1}\boldsymbol{\Upsilon}^{\top}\mathbf{z}$. In this case, for a given $\boldsymbol{v}^* \in \mathbb{R}^{p\times 1}$, the transformed feature map of a particular instance of $\boldsymbol{\gamma}^*$, the predictive posterior distribution

$$\boldsymbol{\beta}^{\top}\boldsymbol{v}^* = f_{\text{post}}(\boldsymbol{\gamma}^*) \sim \mathcal{N}(\mu_{\text{post}}(\boldsymbol{\gamma}^*), \sigma^2_{\text{post}}(\boldsymbol{\gamma}^*))$$

where the mean and variance is defined by:

$$\begin{aligned}\mu_{\text{post}}(\boldsymbol{\gamma}^*) &= \frac{\alpha}{\sigma^2}\mathbf{e}^{\top}\mathbf{L}^{-1}\boldsymbol{v}^* \\ \sigma^2_{\text{post}}(\boldsymbol{\gamma}^*) &= \alpha||\mathbf{L}^{-1}\boldsymbol{v}^*||^2.\end{aligned}$$

It is noted that the computational complexity here scales linearly in the number of observations S and cubically in p.

3.8.2 Warm-starting and acquisition functions

Acquisition functions The exact form of the Expected Improvement (EI) [Močkus, 1975] is defined as follows:

$$\begin{aligned}g(\gamma) &= (\mu_{\text{post}}(\boldsymbol{\gamma}) - z_{\max} - \xi)/\sigma_{\text{post}}(\boldsymbol{\gamma}) \\ \alpha_{\text{EI}}(\boldsymbol{\gamma}; f_{\text{post}}) &= \sigma_{\text{post}}(\boldsymbol{\gamma})(g(\boldsymbol{\gamma})\Phi_{\text{cdf}}(g(\boldsymbol{\gamma})) + \mathcal{N}(g(\boldsymbol{\gamma}); 0, 1)\end{aligned}$$

where $z_{\max}$ refers to the maximum observed z for our *target task*, while Φ_{cdf} and $\mathcal{N}(g(\gamma); 0, 1)$ refers to the CDF and pdf of a standard normal distribution. For experiments, we set the exploration parameter to be $\xi = 0.01$. It should be noted in the case, where the $\alpha_{\text{EI}} \approx 0$ (or numerically close to 0) for all attempted locations, we will use the upper confidence bound (with $\kappa = 2.58$) [Srinivas et al., 2009] instead. To maximise the acquisition function, we first randomly select $300,000$ hyperparameters for evaluation (computationally cheap), to find the top 10 optimum. Initialising from these top 10 hyperparameters, a L-BFGS-B algorithm (computationally expensive) is used to maximise the acquisition function, to select the next hyperparameter for evaluation.

Warm-starting Instead of using the LCB acquisition function (for the first evaluation), an alternative approach is to warm-start [Gomes et al., 2012; Reif et al., 2012; Feurer et al., 2015] based on *learnt* similarities with previous source tasks. For the GP case, we will optimise the marginal likelihood based on all observations from the source tasks, learning the

task similarity function $k_P([\psi(D_1), N_1], [\psi(D_2), N_2])$. As the output domain of k_P lies in $[0, 1]$, we can compute the top M source tasks most similar with our target task. Given this selection, we can extract the best m previous best hyperparameters from each of these source tasks, enabling Mm hyperparameters as warm-start initialisations for our algorithm. For the BLR case, as a joint space over $\boldsymbol{\theta}$, $\psi(D)$ and N is considered, a direct task similarity function is no longer available. Instead we opt for a different approach and extract m previous best hyperparameters from all source tasks, and consider only these hyperparameters for the maximisation of the LCB/EI acquisition function. In practice, we recommend to warm-start with as few evaluations as possible, as:

- Source tasks can be dissimilar to our target task.
- Warm-start hyperparameters may be similar to each other.
- More evaluations are needed before the proposed algorithm can begin to utilise all seen evaluations to explore/exploit for our target task.

3.8.3 Additional experimental details

With the exception of the hyperparameter in the unsupervised toy and the protein random forest example, all other hyperparameters are optimised in the log-scale. In addition, we standardise hyperparameters to have mean 0 and variance 1, when passing them to the GP and BLR, to ensure parameters initialisation are well-defined. Here we provide additional details for our experiments in Section 3.6.

3.8.3.1 Comparison between joint and concatenation embeddings for regression

Here we display additional graphs (Figure 3.10 and 3.11) comparing the embedding of the joint distribution versus the embedding of the conditional distribution and marginal distribution before concatenation. We denote these correspondingly by distGP-joint, distBLR-joint and distGP-concat, distGP-concat. Overall, we observe that their performance is similar.

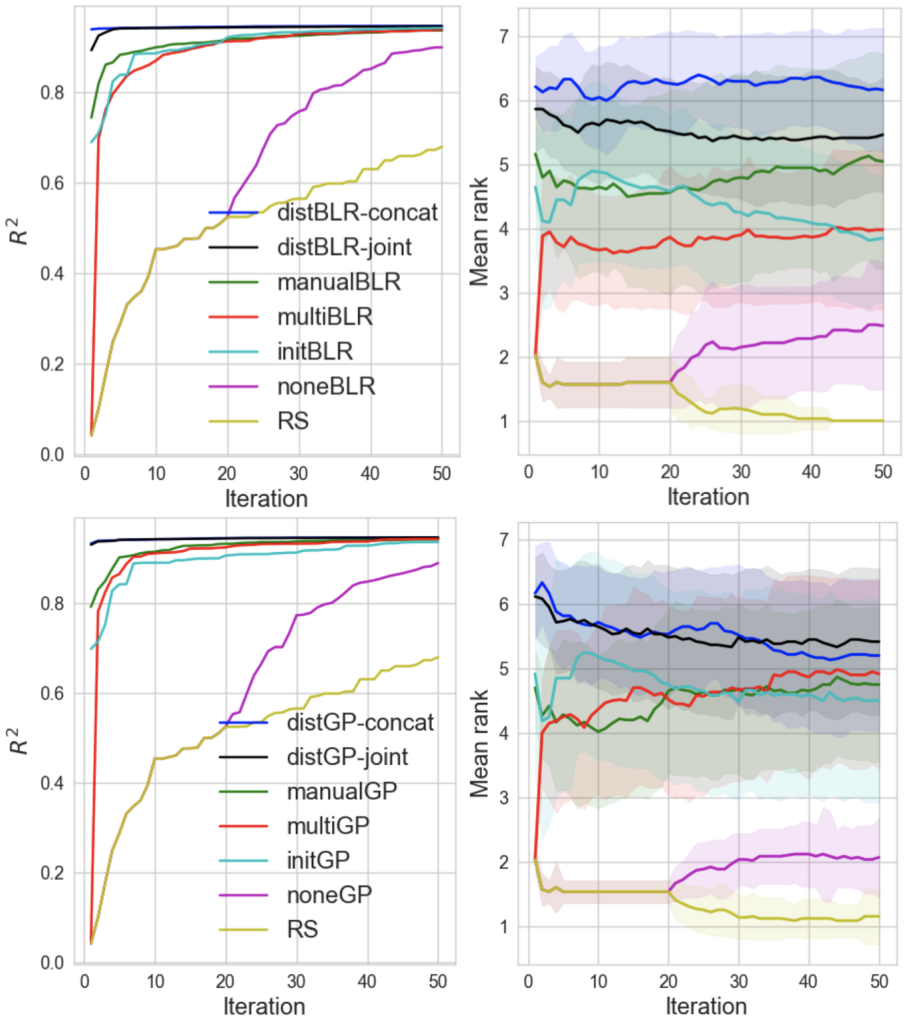

Figure 3.10: Manual meta-features counterexample with 50 iterations (including any initialisation). Here, BLR methods are displayed on the top, while GP methods are displayed on the bottom. Each evaluation here is averaged over 30 runs. **Left:** *Maximum observed* R^2. **Right:** Mean rank (with respect to each run) of the different methodologies, with ± 1 sample standard deviation.

3.8.3.2 Unsupervised toy example

Hyperparameters: $\theta \in [-8, 8]$

Source task's random and BO iterations: $10, 20$

Target task's noneBO random and BO iterations: $5, 10$

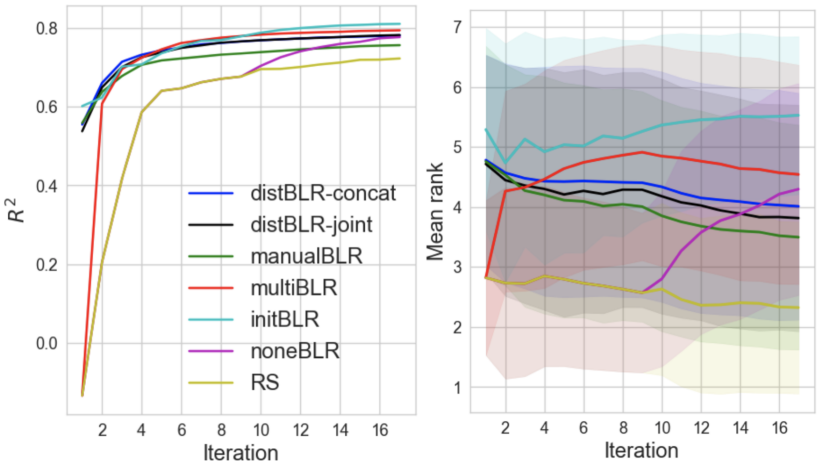

Figure 3.11: Parkinson's experiment with 17 iterations (including any initialisation). Each evaluation here is averaged over 420 runs, with each of the 42 patient set as the target task (repeated for 10 runs) **Left:** *Maximum observed* R^2. **Right:** Mean rank (with respect to each run) of the different methodologies, with ± 1 sample standard deviation.

3.8.3.3 Classification: Similar and not similar source tasks

Hyperparameters: $C \in [2.0^{-7}, 2.0^{10}], \sigma_j \in [2.0^{-3}, 2.0^{5}]$

Source task's random and BO iterations: $75, 75$

Target task's noneBO random and BO iterations: $25, 75$

To generate $\{\mathbf{x}_i^a, y_i^a\}_{i=1}^{N_a}$, we first simulate $\mathbf{x}_i^a \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_6)$. Then in order to sample from the model of an ARD kernel logistic regression, we define an underlying true bandwidth $\tilde{\boldsymbol{\sigma}}^a = [\tilde{\sigma}_1^a, \ldots, \tilde{\sigma}_6^a]$ and use random Fourier features (RFF) [Rahimi & Recht, 2007] to approximate an ARD kernel (with $D = 200$ frequencies) as follows:

$$\boldsymbol{\varphi}_i^a = \sqrt{2/D}\cos(\mathbf{U}\tilde{\mathbf{x}}_i^a + \mathbf{b}) \qquad \mathbf{U} \in \mathbb{R}^{D\times 6}, \mathbf{b} \in \mathbb{R}^D$$

where $\tilde{\mathbf{x}}_i^a = \mathbf{x}_i^a / \tilde{\boldsymbol{\sigma}}^a$ denotes element-wise division by the bandwidths in respective dimensions and $\mathbf{U}_{mn} \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$ and $\mathbf{b}_m \overset{i.i.d.}{\sim} \text{Unif}([0, 2\pi])$. Letting $\boldsymbol{\Phi}^a = [\boldsymbol{\varphi}_1^a, \ldots \boldsymbol{\varphi}_{N_a}^a]^\top$, we let $\tilde{\mathbf{g}}^a = \boldsymbol{\Phi}^a \boldsymbol{\beta}^a$, where $\boldsymbol{\beta}^a \sim \mathcal{N}(0, \mathbf{I}_D)$. We then normalise $\tilde{\mathbf{g}}^a$ to be in the range $[-6, 6]$ and then transform it through the logistic link:

$$p_i^a = \frac{1}{1 + \exp(-\tilde{g}_i^a)}$$

obtaining $p_i^a = P(y_i^a = 1 | \mathbf{x}_i^a)$, using which we can draw a binary output $y_i^a \sim \text{Bernoulli}(p_i^a)$.

3.8.3.4 Regression: Parkinson's dataset

Hyperparameters: $\alpha \in [10.0^{-10}, 0.1], \sigma_j \in [2.0^{-7}, 2.0^{5}]$

Source task's random and BO iterations: $10, 20$

Target task's noneBO random and BO iterations: $9, 8$

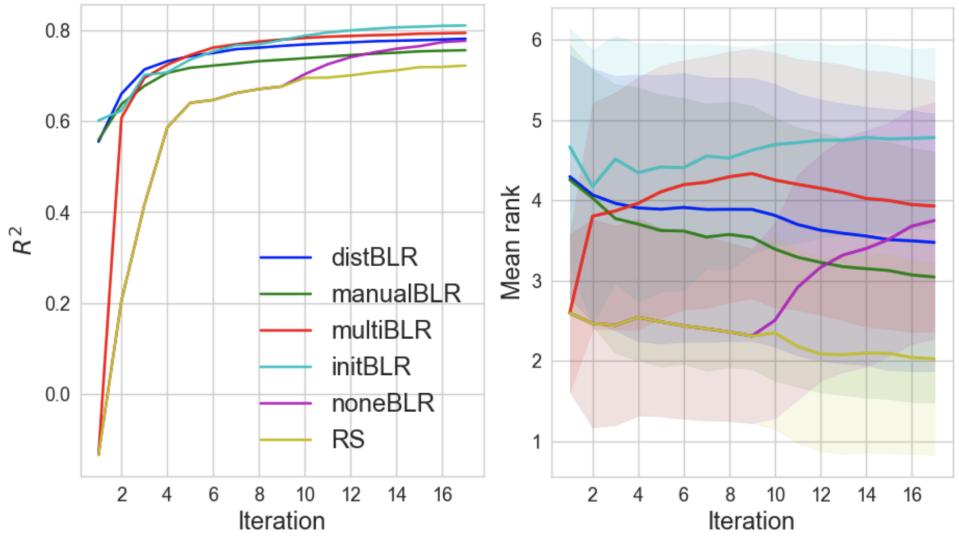

Figure 3.12: Parkinson's experiment with 17 iterations (including any initialisation). Each evaluation here is averaged over 420 runs, with each of the 42 patient set as the target task (repeated for 10 runs) **Left:** *Maximum observed* R^2. **Right:** Mean rank (with respect to each run) of the different methodologies, with ± 1 sample standard deviation.

The Parkinson's disease telemonitoring dataset[9] consists of voice measurements using a telemonitoring device for 42 patients with Parkinson disease (approximately 150 recordings $\in \mathbb{R}^{17}$ each). The label is the clinician's Parkinson disease symptom score for *each recording*. Following a setup similar to Blanchard et al. [2017], we can treat each patient as a separate regression task. In this experiment, in order to allow for comprehensive benchmark comparisons, we consider f which is not prohibitively expensive (hence the problem does not necessarily benefit computationally from Bayesian optimisation). Namely, we employ RBF kernel ridge regression (with hyperparameters α, γ), with f as the coefficient of determination (R^2). In this experiment, we will designate each patient as the target task, while using the other $n = 41$ patients as source tasks. In particular, we will take $s_a = 30$, and hence $S = 1230$, and again since the total number of evaluations is large, will focus on BLR. The results obtained by averaging over different patients as the target task (20 runs per task)

[9] http://archive.ics.uci.edu/ml/datasets/Parkinsons+Telemonitoring

are shown in Figure 3.12. On this dataset, we observe similar behaviour of transfer methods which were able to leverage the source task information and for many patients few-shot the optimum. This suggests the presence of similar source tasks in practice and that this similarity can be exploited in the context of hyperparameter learning.

3.8.3.5 Classification: Protein dataset

Jaccard kernel C-SVM

Hyperparameters: $C \in [2.0^{-7}, 2.0^{10}]$

Source task's random and BO iterations: $10, 10$

Target task's noneBO random and BO iterations: $9, 11$

To compute the Jaccard kernel [Bouchard et al., 2013; Ralaivola et al., 2005], we use of the python package *SciPy*[10] [Jones et al., 2001–] to compute the Jaccard distance, before performing a one subtract each entry to get a similarity matrix. Results are shown in Figure 3.8.

Random Forest

Hyperparameters:

Number of trees: $n\_trees \in \{1, \ldots, 200\}$

Max depth of the tree: $max\_depth \in \{1, \ldots, 32\}$

Min samples required to split a node (after multiplied with N_a):

$min\_samples\_split \in [0.01, 1.0]$

Min samples required at a leaf node (after multiplied with N_a):

$min\_samples\_leaf \in [0.01, 0.5]$

Source task's random and BO iterations: $10, 10$

Target task's noneBO random and BO iterations: $9, 11$

Since n_trees and max_depth are discrete hyperparameters, in practice we round up to the nearest integer, after a continuous version of it is proposed. For additional information on these hyperparameters, please refer to the *RandomForestClassifier*[11] in the Python package *scikit-learn* [Pedregosa et al., 2011]. Results are shown in Figure 3.9.

[10] https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cdist.html

[11] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Chapter 4

Testing and Learning on Distributions with Symmetric Noise Invariance

This chapter is based on the following paper:

Ho Chung Leon Law, Christopher Yau, and Dino Sejdinovic.
Testing and learning on distributions with symmetric noise invariance [Law et al., 2017]
Advances in Neural Information Processing Systems (NeurIPS), 2017.

Kernel embeddings of distributions and the Maximum Mean Discrepancy (MMD), the resulting distance between distributions, are useful tools for fully nonparametric two-sample testing and learning on distributions. However, it is rare that all possible differences between samples are of interest – discovered differences can be due to different types of measurement noise, data collection artefacts or other irrelevant sources of variability. We propose distances between distributions which encode invariance to additive symmetric noise, aimed at testing whether the assumed true underlying processes differ. Moreover, we construct invariant features of distributions, leading to learning algorithms robust to the impairment of the input distributions with symmetric additive noise.

4.1 Introduction

There are many sources of variability in data, and not all of them are pertinent to the questions that a data analyst may be interested in. Consider, for example, a nonparametric two-sample testing problem, which has recently been attracting significant research interest, especially

in the context of kernel embeddings of distributions [Chwialkowski et al., 2015; Gretton et al., 2012a; Jitkrittum et al., 2016]. We observe samples $\{\mathbf{x}_i\}_{i=1}^{N_1}$ and $\{\mathbf{y}_i\}_{i=1}^{N_2}$ from two data generating processes P_X and P_Y, respectively, and would like to test the null hypothesis that $P_X \stackrel{d}{=} P_Y$ without making any parametric assumptions on these distributions. With a large sample-size, the minutiae of the two data generating processes are uncovered (e.g. slightly different calibration of the data collecting equipment, different numerical precision), and we ultimately reject the null hypothesis, even if the sources of variation across the two samples may be irrelevant for the analysis.

Similarly, we may be interested in *learning on distributions* [Muandet et al., 2012; Sutherland et al., 2016; Szabó et al., 2016], where the appropriate level of granularity in the data is distributional. For example, each label y^a in supervised learning is associated to a whole bag of observations $B_a = \{\mathbf{x}_i^a\}_{i=1}^{N_a}$ – assumed to come from a probability distribution P_a, or we may be interested in clustering such bags of observations. Again, nonparametric distances used in such contexts to facilitate a learning algorithm on distributions, such as Maximum Mean Discrepancy (MMD) [Gretton et al., 2012a], can be sensitive to irrelevant sources of variation and may lead to suboptimal or even misleading results, in which case building predictors which are invariant to noise is of interest.

While it may be tempting to revert back to a parametric setup and work with simple, easy to interpret models, we argue that a different approach is possible: we stay within a nonparametric framework, exploit the irregular and complicated nature of real life distributions and *encode invariances* to sources of variation assumed to be irrelevant. In this contribution, we focus on *invariances to symmetric additive noise* on each of the data generating distributions. Namely, assume that the a-th sample $\{\mathbf{x}_i^a\}_{i=1}^{N_a}$ we observe does not follow the distribution P_a of interest but instead its convolution $P_a \star \mathcal{E}_a$ with some unknown noise distributions $\mathcal{E}_a$ assumed to be symmetric about 0 (we also require that it has a positive characteristic function). We would like to assess the differences between P_a and $P_{a'}$ while allowing $\mathcal{E}_a$ and $\mathcal{E}_{a'}$ to differ in an arbitrary way. We investigate two approaches to this problem: (1) measuring the degree of asymmetry of the paired differences $\{\mathbf{x}_i^a - \mathbf{x}_i^{a'}\}$, and (2) comparing the *phase functions* of the corresponding samples. While the first approach is simpler and presents a sensible solution for the two-sample testing problem, we demonstrate that phase functions

give a much better gauge on the *relative comparisons* between bags of observations, as required for learning on distributions.

4.2 Background

We will say that a random vector E on $\mathbb{R}^d$ is a *symmetric positive definite (SPD) component* if its characteristic function is positive, i.e. $\varphi_E(\boldsymbol{\omega}) = \mathbb{E}_E\left[\exp(i\boldsymbol{\omega}^\top E)\right] > 0, \forall \boldsymbol{\omega} \in \mathbb{R}^d$. This means that E is (1) symmetric about zero, i.e. E and $-E$ have the same distribution and (2) if it has a density, this density must be a positive definite function [Rossberg, 1995]. Note that many distributions used to model additive noise, including the spherical zero-mean Gaussian distribution, as well as multivariate Laplace, Cauchy or Student's t (but not uniform), are all SPD components.

Following the terminology similar to that of Delaigle & Hall [2016], we will say that a random vector X on $\mathbb{R}^d$ is *decomposable* if its characteristic function can be written as $\varphi_X = \varphi_{X_0}\varphi_E$, with $\varphi_E > 0$. Thus, if X can be written in the form $X = X_0 + E$, where X_0 and E are independent and E is an SPD noise component, then X is decomposable. We will say that X is *indecomposable* if it is not decomposable. In this chapter, we will assume that mostly the indecomposable components of distributions are of interest and we will construct tools to directly measure differences between these indecomposable components, encoding invariance to other sources of variability. The class of Borel probability measures on $\mathbb{R}^d$ will be denoted $\mathcal{M}_+^1(\mathbb{R}^d)$, while the class of indecomposable probability measures will be denoted by $\mathcal{I}(\mathbb{R}^d) \subseteq \mathcal{M}_+^1(\mathbb{R}^d)$.

For shift-invariant kernels k on $\mathbb{R}^d$, using Bochner's characterisation [Wendland, 2004, 6.2], the squared MMD (overview in Section 1.2.1.3 in Chapter 1) can be written as a weighted L_2-distance between characteristic functions [Sriperumbudur et al., 2010, Corollary 4]

$$\|\mu_{P_X} - \mu_{P_Y}\|_{\mathcal{H}_k}^2 = \int_{\mathbb{R}^d} \left|\varphi_{P_X}(\boldsymbol{\omega}) - \varphi_{P_Y}(\boldsymbol{\omega})\right|^2 d\Lambda(\boldsymbol{\omega}), \tag{4.1}$$

where Λ is the non-negative spectral measure (inverse Fourier transform) of kernel k as a function of $\mathbf{x} - \mathbf{y}$, while $\varphi_{P_X}(\omega)$ and $\varphi_{P_Y}(\omega)$ are the characteristic functions of probability measures P_X and P_Y.

4.3 Phase Discrepancy and Phase features

While MMD and kernel embeddings are related to characteristic functions, and indeed the same connection forms a basis for fast approximations to kernel methods using random Fourier features [Rahimi & Recht, 2007] (overview in Section 1.2.1 in Chapter 1), the relevant notion in our context is the *phase function* of a probability measure, recently used for nonparametric deconvolution by Delaigle & Hall [2016]. In this section, we overview this formalism. Based on the empirical phase functions, we will then derive and investigate hypothesis testing and learning framework using *phase features of distributions*.

In nonparametric deconvolution [Delaigle & Hall, 2016], the goal is to estimate the density function f_0 of a univariate r.v. X_0, but in general we only have noisy samples $X_1, \ldots, X_N \overset{iid}{\sim} X = X_0 + E$, where E denotes an independent noise term. Even though the distribution of E is unknown, making the assumption that E is an SPD noise component, and that X_0 is indecomposable, i.e. X_0 itself does not contain any SPD noise components, Delaigle & Hall [2016] show that it is possible to obtain consistent estimates of f_0.

They distinguish between the symmetric noise and the underlying indecomposable component by matching phase functions, defined as

$$\rho_X(\boldsymbol{\omega}) = \frac{\varphi_X(\boldsymbol{\omega})}{|\varphi_X(\boldsymbol{\omega})|} \tag{4.2}$$

where $\varphi_X(\boldsymbol{\omega})$ denotes the characteristic function of X. Observe that $|\rho_X(\boldsymbol{\omega})| = 1$, and thus we are effectively removing the amplitude information from the characteristic function. For a SPD noise component E, the phase function is $\rho_E(\boldsymbol{\omega}) \equiv 1$. But then since $\varphi_X = \varphi_{X_0}\varphi_E$, we have that $\rho_{X_0} = \rho_X = \varphi_X/|\varphi_X|$, i.e. the phase function is invariant to additive SPD noise components. This motivates us to construct explicit feature maps of distributions with the same property and similarly to the motivation of Delaigle & Hall [2016], we argue that real-world distributions of interest often exhibit certain amount of irregularity and it is exactly this irregularity which is exploited in our methodology.

In analogy to the MMD, we first define the phase discrepancy (PhD) as a weighted L_2-distances between the phase functions:

$$\mathrm{PhD}(X, Y) = \int_{\mathbb{R}^d} |\rho_X(\boldsymbol{\omega}) - \rho_Y(\boldsymbol{\omega})|^2 \, d\Lambda(\boldsymbol{\omega}) \tag{4.3}$$

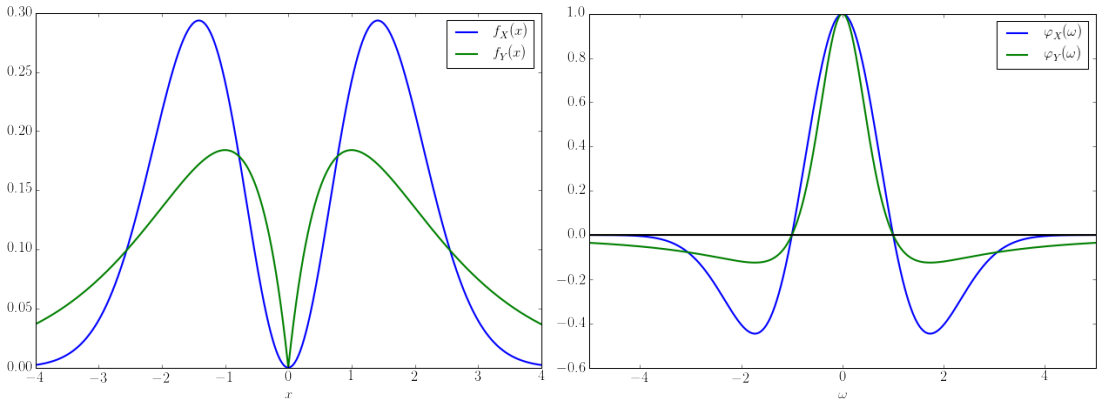

Figure 4.1: Example of two indecomposable distributions which have the same phase function. **Left**: densities. **Right**: characteristic functions.

for some non-negative measure Λ (w.l.o.g. a probability measure). Now suppose we write $X = X_0 + U$, $Y = Y_0 + V$, where U and V are SPD noise components. This then implies $\rho_X = \rho_{X_0}$ and $\rho_Y = \rho_{Y_0}$ Λ-everywhere, so that $\mathrm{PhD}(X, Y) = \mathrm{PhD}(X_0, Y_0)$. It is clear then that the PhD is not affected by additive SPD noise components, so it captures desired invariance. However, the PhD for Λ supported everywhere is in fact not a proper metric on the indecomposable probability measures $\mathcal{I}(\mathbb{R}^d)$, as one can find indecomposable random variables X and Y s.t. $\rho_X = \rho_Y$ and thus $\mathrm{PhD}(X, Y) = 0$. We now present such an example. Let X and Y be (univariate) random variables with densities

$$f_X(x) = \frac{1}{\sqrt{2\pi}} x^2 \exp(-x^2/2), \quad f_Y(x) = \frac{1}{2}|x| \exp(-|x|).$$

Then it can be directly checked that their characteristic functions are given by

$$\varphi_X(\omega) = (1 - \omega^2) \exp(-\omega^2/2), \quad \varphi_Y(\omega) = \frac{1 - \omega^2}{(1 + \omega^2)^2}.$$

Thus, the phase functions coincide and are equal to

$$\rho_X(\omega) = \rho_Y(\omega) = \begin{cases} +1, |\omega| < 1, \\ -1, |\omega| > 1, \\ \text{undefined}, \omega \in \{-1, 1\}. \end{cases}$$

Further, it can be checked that even though they are symmetric, X and Y are indecomposable, cf. Linnik & Ostrovskii [1977], which use a related but distinct notion of indecomposability of random variables. The plots of the densities and characteristic functions of X and Y are given in Figure 4.1. While such cases appear contrived, we hence restrict attention

to a subset of indecomposable probability measures $\mathcal{P}(\mathbb{R}^d) \subset \mathcal{I}(\mathbb{R}^d)$, which are uniquely determined by phase functions, i.e. $\forall P, Q \in \mathcal{P}(\mathbb{R}^d) : \rho_P = \rho_Q \Rightarrow P = Q$.

We now have the two following propositions (proofs are given in Appendix 4.7.1).

Proposition 1.

$$\mathit{PhD}(X, Y) = 2 - 2 \int \left(\frac{\mathbb{E}\xi_{\boldsymbol{\omega}}(X)}{\|\mathbb{E}\xi_{\boldsymbol{\omega}}(X)\|} \right)^{\top} \left(\frac{\mathbb{E}\xi_{\boldsymbol{\omega}}(Y)}{\|\mathbb{E}\xi_{\boldsymbol{\omega}}(Y)\|} \right) d\Lambda(\boldsymbol{\omega})$$

where $\xi_{\omega}(\mathbf{x}) = \left[\cos\left(\boldsymbol{\omega}^{\top}\mathbf{x}\right), \sin\left(\boldsymbol{\omega}^{\top}\mathbf{x}\right)\right]^{\top}$ *and* $\|\cdot\|$ *denotes the standard* L_2 *norm.*

Proposition 2.

$$K(P_X, P_Y) = \int \left(\frac{\mathbb{E}\xi_{\boldsymbol{\omega}}(X)}{\|\mathbb{E}\xi_{\boldsymbol{\omega}}(X)\|} \right)^{\top} \left(\frac{\mathbb{E}\xi_{\boldsymbol{\omega}}(Y)}{\|\mathbb{E}\xi_{\boldsymbol{\omega}}(Y)\|} \right) d\Lambda(\boldsymbol{\omega})$$

is a positive definite kernel on probability measures.

Now, we can construct an approximate explicit feature map for kernel K. Taking a sample $\{\boldsymbol{\omega}_j\}_{j=1}^{J} \sim \Lambda$, we define $\Psi : P_X \mapsto \mathbb{R}^{2J}$ given by $\Psi(P_X) = \sqrt{\frac{1}{J}} \left[\frac{\mathbb{E}\xi_{\boldsymbol{\omega}_1}(X)}{\left\|\mathbb{E}\xi_{\boldsymbol{\omega}_1}(X)\right\|}, \ldots, \frac{\mathbb{E}\xi_{\boldsymbol{\omega}_J}(X)}{\left\|\mathbb{E}\xi_{\boldsymbol{\omega}_J}(X)\right\|} \right]$. We will refer to $\Psi(\cdot)$ as the *phase features*. Note that these are very similar to the mean embedding of the random Fourier feature map[1]:

$$\Phi(P) = \sqrt{\frac{1}{J}} \left[\mathbb{E}\xi_{\boldsymbol{\omega}_1}(X), \ldots, \mathbb{E}\xi_{\boldsymbol{\omega}_J}(X)\right] \in \mathbb{R}^{2J} \tag{4.4}$$

but here the $\cos, \sin$-pair corresponding to each frequency is normalised to have unit L_2 norm. In other words, $\Psi(\cdot)$ can be thought of as the evaluations of the phase function at selected frequencies. By construction, phase features are invariant to additive SPD noise components. For an empirical measure, we simply have the following:

$$\Psi(\widehat{P}_X) = \sqrt{\tfrac{1}{J}} \left[\frac{\hat{\mathbb{E}}\xi_{\boldsymbol{\omega}_1}(X)}{\left\|\hat{\mathbb{E}}\xi_{\boldsymbol{\omega}_1}(X)\right\|}, \ldots, \frac{\hat{\mathbb{E}}\xi_{\boldsymbol{\omega}_J}(X)}{\left\|\hat{\mathbb{E}}\xi_{\boldsymbol{\omega}_J}(X)\right\|} \right] \tag{4.5}$$

where we have replaced the expectations by their empirical estimates. Because $\left\|\Psi(\widehat{P}_X)\right\| = 1$, we can construct

$$\widehat{\mathrm{PhD}}(\widehat{P}_X, \widehat{P}_Y) = \left\|\Psi(\widehat{P}_X) - \Psi(\widehat{P}_Y)\right\|^2 = 2 - 2\Psi(\widehat{P}_X)^{\top}\Psi(\widehat{P}_Y), \tag{4.6}$$

which is a Monte Carlo estimator of $\mathrm{PhD}(\widehat{P}_X, \widehat{P}_Y)$. In summary, $\Psi(\widehat{P}) \in \mathbb{R}^{2J}$ is an explicit feature vector of the empirical distribution which encodes invariance to additive SPD noise

[1] To be understood as the evaluations (real and complex part stacked together) of the characteristic function φ_P at frequencies $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_J$.

components present in P, empirically demonstrated in Section 4.5.2.1. We also note that unlike the population expression $\Psi(P)$, the empirical estimator $\Psi(\widehat{P})$ will in general have a distribution affected by the noise components and is thus only approximately invariant, but we observe that it captures invariance very well as long as the signal-to-noise regime remains relatively high (Section 4.5.1). Given this feature map, it can now be directly applied to (1) two-sample testing up to SPD components, where the distance between the phase features, i.e. an estimate (4.6) of the PhD, can be used as a test statistic, with details given in Section 4.5.1 and (2) learning on distributions, where we use phase features as the explicit feature map for a bag of samples.

Although we have assumed an indecomposable underlying distribution so far, this assumption is not strict. For distribution regression, if the indecomposable assumption is invalid, given that the underlying distribution is irregular, it may still be useful to encode invariance as long as the benefit of removing the SPD components irrelevant for learning outweighs the signal in the SPD part of the distribution, i.e. there is a trade off between SPD noise and SPD signal. In practice, the phase features we propose can be used to encode such invariance where appropriate or in conjunction with other features which do not encode invariance.

In order to construct the approximate mean embeddings for learning, we first compute an explicit feature map by taking averages of the Fourier features, as given by

$$\Phi(\widehat{P}_X) = \sqrt{\frac{1}{J}} \left[\hat{\mathbb{E}}\xi_{\boldsymbol{\omega}_1}(X), \ldots, \hat{\mathbb{E}}\xi_{\boldsymbol{\omega}_J}(X) \right]. \tag{4.7}$$

For phase features, we need to compute an additional normalisation term over each frequency as in (4.5). To obtain the set of frequencies $\{\boldsymbol{\omega}_j\}_{j=1}^J$, we can draw samples from a probability measure Λ corresponding to an inverse Fourier transform of a shift-invariant kernel, e.g. Gaussian kernel. However, given a supervised signal, we can also optimise a set of frequencies $\{\boldsymbol{\omega}_j\}_{j=1}^J$ that will give us a useful representation and good discriminative performance. In other words, we no longer focus on a specific shift-invariant kernel k, but are *learning discriminative Fourier/phase features*. To do this, we can construct a neural network (NN) with special activation functions, pooling layers as shown in Algorithm 1 and Figure 4.10 in Appendix 4.7.3.

4.4 Asymmetry in paired differences

We now consider a separate approach to nonparametric two-sample test, where we wish to test the null hypothesis that $H_0 : P_{X_0} \stackrel{d}{=} P_{Y_0}$ vs. the general alternative, but we only have iid samples arising from $X \sim P_{X_0} \star \mathcal{E}_1$ and $Y \sim P_{Y_0} \star \mathcal{E}_2$. i.e.

$$X = X_0 + U \quad Y = Y_0 + V$$

where $X_0 \sim P_{X_0}$, $Y_0 \sim P_{Y_0}$ lie in the space of $\mathcal{P}(\mathbb{R}^d)$ of indecomposable distributions uniquely determined by phase functions and U and V are SPD noise components. With this setting, we have the following proposition (proof in Appendix 4.7.1):

Proposition 3. *Under the null hypothesis H_0, $X - Y$ is SPD* $\iff$ $X_0 \stackrel{d}{=} Y_0$.

This motivates us to simply perform a two-sample test on $X - Y$ and $Y - X$ since its rejection would imply rejection of $X_0 \stackrel{d}{=} Y_0$, as it tests for symmetry. However, note that this is a test for symmetry only and that for consistency against all alternatives, positivity of characteristic function would need to be checked separately. Now, given two iid samples $\{\mathbf{x}_i\}_{i=1}^N$ and $\{\mathbf{y}_i\}_{i=1}^N$ with N even, we split the two samples into two halves and compute $\mathbf{z}_i^1 = \mathbf{x}_i - \mathbf{y}_i$ on one half and $\mathbf{z}_i^2 = \mathbf{y}_i - \mathbf{x}_i$ on the other half, and perform a nonparametric two-sample test on Z_1 and Z_2 (which are, by construction, independent of each other). The advantage of this regime is that we can use any two-sample test – in particular in this chapter, we will focus on the *linear time* mean embedding (ME) test [Jitkrittum et al., 2016] (overview in Section 1.2.1.3 of Chapter 1), which was found to have performance similar to or better than the original MMD two-sample test [Gretton et al., 2012a], and explicitly formulates a criterion which maximises the test power. We will refer to the resulting test on paired differences as the Symmetric Mean Embedding (SME).

Although we have assumed here that X_0, Y_0 lie in the space $\mathcal{P}(\mathbb{R}^d)$ of indecomposable distributions, in practice, the SME test would not reject if the underlying distributions of interest *differ only in the symmetric components* (or in the SPD components for the PhD test). We argue this to be unlikely due to real life distributions being complex in nature with interesting differences often having a degree of asymmetry. In practice, we recommend the use of the

ME and SME or PhD test together to provide an exploratory tool to understand the underlying differences, as demonstrated in the Higgs Data experiment in Section 4.5.1.2. It is tempting to also consider learning on distributions with invariances using this formalism. However note that the MMD on paired differences is *not invariant to the additive SPD noise components* under the alternative, i.e. in general $\text{MMD}(X-Y, Y-X) \neq \text{MMD}(X_0-Y_0, Y_0-X_0)$. This means that the paired differences approach to learning is sensitive to the actual type and scale of the additive SPD noise components, hence not suitable for learning. The mathematical details and empirical experiments to show this are correspondingly presented in Appendix 4.7.2 and Section 4.5.2.1.

4.5 Experiments

4.5.1 Two-sample tests with invariances

In this section, we demonstrate the performance of the SME test and the PhD test on both artificial and real-world data for testing the hypothesis $H_0 : X_0 \stackrel{d}{=} Y_0$ based on samples $\{\mathbf{x}_i\}_{i=1}^N$ from $X_0 + U$ and $\{\mathbf{y}_i\}_{i=1}^N$ from $Y_0 + V$, where U and V are arbitrary SPD noise components (we assume the same number of samples for simplicity). SME test follows the setup in Jitkrittum et al. [2016] but applied to $\{\mathbf{x}_i - \mathbf{y}_i\}_{i=1}^{N/2}$ and $\{\mathbf{y}_i - \mathbf{x}_i\}_{i=N/2+1}^N$. For the PhD test, we use the test statistic $\widehat{\text{PhD}}(\hat{P}_X, \hat{P}_Y)$ of (4.3). It is unclear what the exact form of the null distribution is, so we use a permutation test, by recomputing this statistic on the samples which are first merged and then randomly split in the original proportions. While we are combining samples with different distributions, the permutation test is still justified since, under the null hypothesis $X_0 \stackrel{d}{=} Y_0$, the resulting characteristic function φ_{null} of the mixture can be written as

$$\varphi_{null} = \frac{1}{2}\varphi_{X_0}\varphi_U + \frac{1}{2}\varphi_{X_0}\varphi_V = \varphi_{X_0}(\frac{1}{2}\varphi_U + \frac{1}{2}\varphi_V) \tag{4.8}$$

and since the mixture of the SPD noise terms is also SPD, we have that $\rho_{null} = \rho_{X_0} = \rho_{Y_0}$. For our experiments, we denote by N the sample size, d the dimension of the samples, and we take $\alpha = 0.05$ to be the significance level. In the SME test, we take the number of test locations J to be 10, and use 20% of the samples to optimise the test locations. All

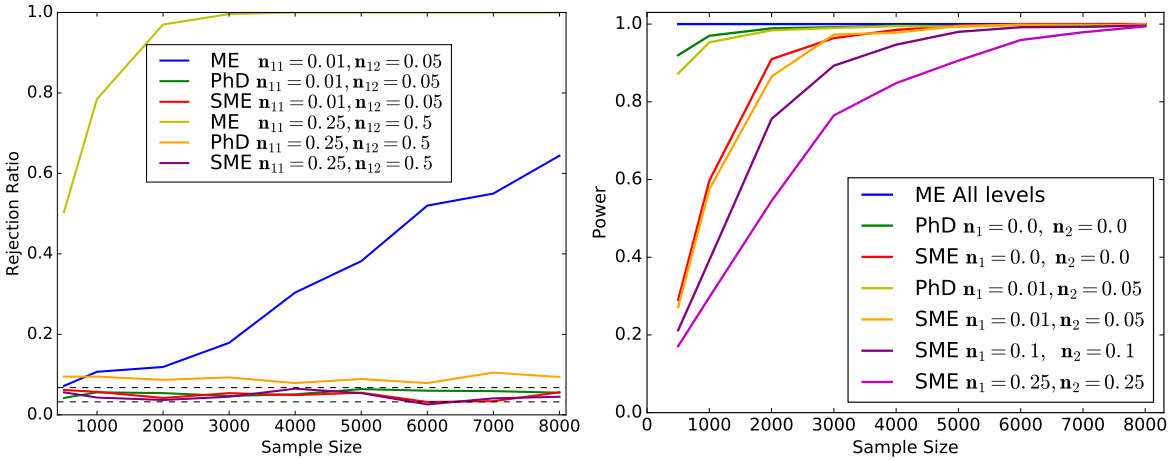

Figure 4.2: Type I error and Power under various additional symmetric noise in the synthetic χ^2 dataset. Dashed line is the 99% Wald interval here. **Left:** Type I error, n_{11} denotes the noise to signal ratio for the first set of samples and n_{12} for the second set. **Right:** Power, n_1 denotes the noise to signal ratio for the X set of samples and n_2 denotes the noise to signal ratio for the Y set of samples.

experimental results are averaged over 1000 runs, where each run repeats the simulation or randomly samples without replacement from the dataset.

4.5.1.1 Synthetic example: Noisy χ^2

We start by demonstrating our tests with invariances on a simulated dataset where X_0 and Y_0 are random vectors with $d = 5$, each dimension is the same in distribution and follows $\chi^2(4)/4$ and $\chi^2(8)/8$ respectively, i.e. χ^2 random variables, with different degrees of freedom, rescaled to have the same mean 1 (but have different variances, $1/2$ and $1/4$ respectively). An illustration of the true and empirical phase and characteristic function with noise for these two distributions can be found in Appendix 4.7.4. We construct samples $\{\mathbf{x}_i^{n_1}\}_{i=1}^N$ and $\{\mathbf{y}_i^{n_2}\}_{i=1}^N$ such that $\mathbf{x}_i^{n_1} \sim X_0 + U$ iid, where $U \sim \mathcal{N}(0, \sigma_1^2 \mathbf{I})$ and similarly $\mathbf{y}_i^{n_2} \sim Y_0 + V$ iid, where $V \sim \mathcal{N}(0, \sigma_2^2 \mathbf{I})$, n_i denotes the noise-to-signal ratio given by the ratio of variances in each dimension, i.e. $n_1 = 2\sigma_1^2$ and $n_2 = 4\sigma_2^2$.

We first verify that the Type I error is indeed controlled at our design level of $\alpha = 0.05$ *up to various additive SPD noise components*. This is shown in Figure 4.2 (left), where $X_0 \overset{d}{=} Y_0$, both constructed using $\chi^2(4)/4$, with the noiseless case found in Figure 4.3 (left). It is noted here that the ME test rejects the null hypothesis for even a small difference in noise levels, hence it is unable to let us *target the underlying distributions* we are concerned with. This is

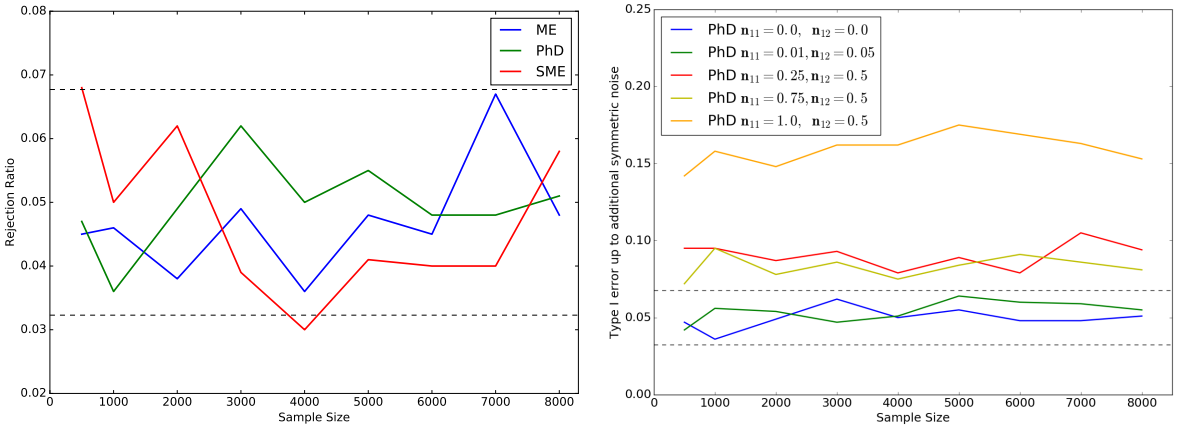

Figure 4.3: Type I error results for the synthetic example with χ^2 **Left:** With no noise added for the ME, PhD and SME test. **Right:** Various additive Gaussian components, our base distribution without addition of noise is $\chi^2(4)/4$. Here n_{11} refers to the noise to signal ratio for the first set of samples and n_{12} refers to the second set of samples.

unlike the SME test which controls the Type I error even for large differences in noise levels. The PhD test, on the other hand, while correctly controlling Type I at small noise levels, was found to have inflated Type I error rates for large noise, as shown in Figure 4.3 (right). This is not surprising, as the null distribution was constructed by using a permutation test, using (4.8) and if the estimated phase features are biased, in the regime with large additive Gaussian noise, then the following may not be true approximately: $\hat{\rho}_{null} = \hat{\rho}_{X_0} = \hat{\rho}_{Y_0}$, leading a to a biased null distribution.

Next, we investigate the power, shown in Figure 4.2 (right). For a fair comparison, we have included the PhD test power only for small noise levels, in which the Type I error is controlled at the design level. In these cases, the PhD test has better power than the SME test. This is not surprising, as for the SME we have to halve the sample size in order to construct a valid test. However, recall that the PhD test has an inflated Type I error for large noises, which means that its results should be considered with caution in practice. ME test rejects at all levels at all sample sizes as it picks up all possible differences. SME and PhD are by construction more conservative tests whose rejection provides a much stronger statement: two samples differ even when *all arbitrary additive SPD components* have been stripped off.

In practice, if it is subtle effects we are looking for, with larger samples, we recommend the use of the SME test, however if this is not the case, then the PhD test is more appropriate,

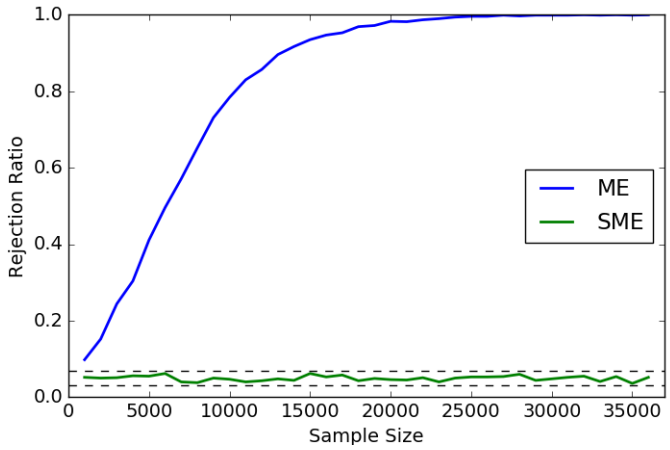

Figure 4.4: Rejection ratio vs. sample size for extremely low level features for Higgs dataset. Dashed line is the 99% Wald interval for 1000 repetitions for $\alpha = 0.05$. Note PhD is not used here, due to its expensive computational cost.

Table 4.1: Power for various sample size for high level features of the Higgs dataset

SAMPLE SIZE N	SME POWER	ME POWER
500	0.94	1.0
600	0.969	0.999
700	0.987	1.0
800	0.989	1.0
900	0.994	1.0
1000	0.995	1.0

as it has good power for low sample size. In fact, the PhD test has power comparable with that of the ME test, however it is cautioned that it does not control the Type I error for larger additional SPD differences and requires more computational power.

4.5.1.2 Higgs dataset

The UCI Higgs dataset [Baldi et al., 2014; Lichman, 2013] is a dataset with 11 million observations, where the problem is to distinguish between the signal process where Higgs bosons are found, versus the background process that do not produce Higgs bosons. In particular, we will consider a two-sample test with the ME and SME test on the high level features derived by physicists, as well as a two-sample test on four extremely low level features (azimuthal angular momentum ϕ measured by four particle jets in the detector). The high level features here (in $\mathbb{R}^7$) have been shown to have good discriminative properties in Baldi et al. [2014]. Thus, we expect them to have different distributions across two processes. Denoting

by X the high level features of the process without Higgs Boson, and Y as the corresponding distribution for the processes where Higgs bosons are produced, we test the null hypothesis that the indecomposable parts of X and Y agree. The results can be found in Table 4.1, which shows that the high level features differ even up to additive SPD components, with a high power for the SME and ME test even at small sample sizes. Now we perform the same experiment, but with the low level features $\in \mathbb{R}^4$, commented in Baldi et al. [2014] to carry very little discriminating information, using the setup from Chwialkowski et al. [2015].

The results for the ME and SME test can be found in Figure 4.4. Here we observe that while ME test clearly rejects and finds the difference between the two distributions, there is no evidence that the indecomposable parts of the joint distributions of the angular momentum actually differ. In fact, the test rejection rate remains around the chosen design level of $\alpha = 0.05$ for all sample sizes. This highlights the significance in using the SME test, suggesting that the nature of the difference between the two processes can be potentially explained by some additive symmetric noise components which may be irrelevant for discrimination, providing an insight into the dataset. Furthermore, this also highlights the argument that given two samples from complex data collection and generation processes, a nonparametric two-sample test like ME will likely reject given sufficient sample sizes, even if the discovered difference may not be of interest. With the SME test however, we can ask a much more subtle question about the differences between the assumed true underlying processes. Figures showing that the Type I error is controlled at the design level of $\alpha = 0.05$ for both low and high level features can be found in Figure 4.5. Here the null hypothesis is true, as we only consider samples drawn from Y, corresponding to the distribution of the processes where the Higgs Boson are produced.

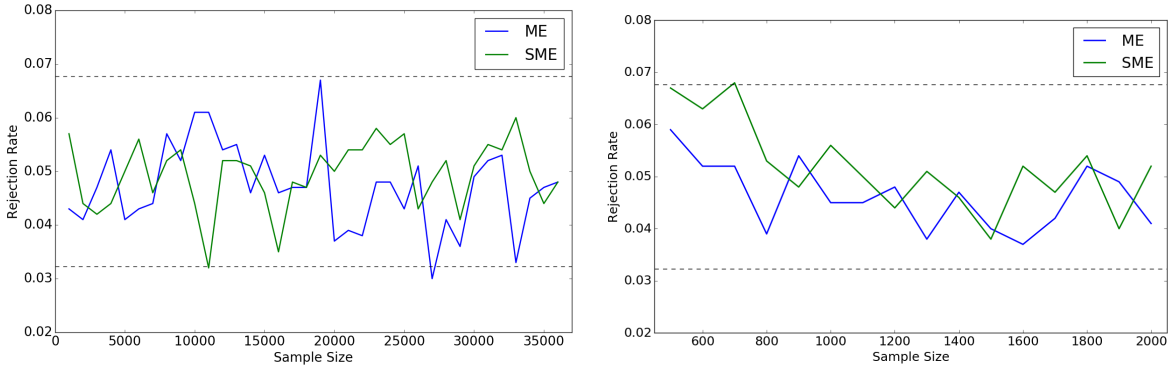

Figure 4.5: Type I error for the Higgs Dataset. **Left:** Extremely low level features **Right:** High level features. The black dashed line is the 99% Wald interval $\alpha \pm 2.57\sqrt{\alpha(1-\alpha)/1000}$, where here $\alpha = 0.05$ is the significance level and 1000 is the number of repetitions.

4.5.2 Learning with Phase features

4.5.2.1 Demonstration that MMD on paired differences is not invariant to SPD noise

Here, we demonstrate empirically[2] that MMD on paired differences is not suitable for learning, as $\text{MMD}(X - Y, Y - X) \neq \text{MMD}(X_0 - Y_0, Y_0 - X_0)$ under the alternative hypothesis. Using the synthetic experimental setup as before, we simulate 100 noiseless bags from two scaled χ^2 distributions $X_0 \sim \chi^2(4)/4$ and $Y_0 \sim \chi^2(8)/8$, where each bag contains 1000 samples. We add varying levels of Gaussian noise to each bag, i.e. the bags are of the form $X_a = X_0 + \mathcal{N}(0, Z_a)$ and $Y_a = Y_0 + \mathcal{N}(0, W_a)$, where $Z_a, W_a \sim U[0, 0.1]$. We compute the estimate of the MMD on paired differences, the squared distance between Fourier features (an estimate of MMD) and the squared distance between phase features (an estimate of PhD) for all pairs of bags. In all computations, we used the same set of frequencies $\{\boldsymbol{\omega}_j\}_{j=1}^{100}$ (sampled from a Gaussian distribution). We do the same for the noiseless samples (or use analytic expressions where available).

The results are shown in Figure 4.6. We see that the MMD on paired differences is not invariant to SPD noise components (the noiseless case is indicated by the red line). This is unlike the phase features, which maintain some level of invariance, the estimates stay away from 0 – preserving the signal about the difference of indecomposable χ^2 components – and the mode is nearer the true value, even though there is clearly some variance, however this is

[2] Additional mathematical details are presented in Appendix 4.7.2.

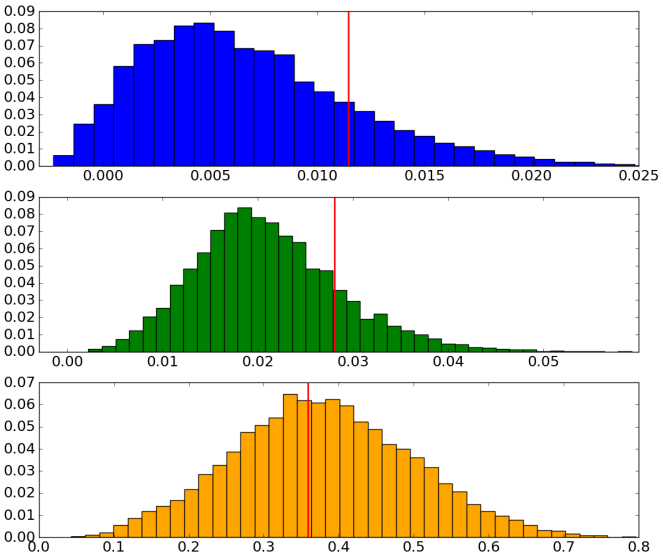

Figure 4.6: Histograms on various estimates for all pairs of bags with varying additive noise, red line denotes the noiseless case. **Top:** Estimated MMD on paired differences for all pair of bags, the red line given by the mean of the estimated MMD on paired differences for bags without noise. **Middle:** Squared distance between Fourier features (an estimate of MMD). **Bottom:** Squared distance between phase features (an estimate of PhD).

expected as its PhD population expression is invariant, but not its estimator, furthermore the frequencies are sampled (with the median heuristic bandwidth) and not learnt. This suggests that phase features are more suitable for invariant learning on distributions than MMD on paired differences. The Fourier features are also given for comparison, but these are not expected to be invariant, as shown.

4.5.2.2 Aerosol dataset

To demonstrate the phase features invariance to SPD noise component[3], we use the Aerosol MISR1 dataset also studied by Szabó et al. [2016] and Wang et al. [2012] and consider a situation with *covariate shift* [Quinonero-Candela et al., 2009] on distribution inputs: the testing data is impaired by additive SPD components different to that in the training data. Here, we have an aerosol optical depth (AOD) multi-instance learning problem with 800 bags, where each bag contains 100 randomly selected multi-spectral (potentially cloudy) pixels within 20km radius around an AOD sensor. The label y^a for each bag is given by the

[3]Code is available at https://github.com/hcllaw/phase_learn

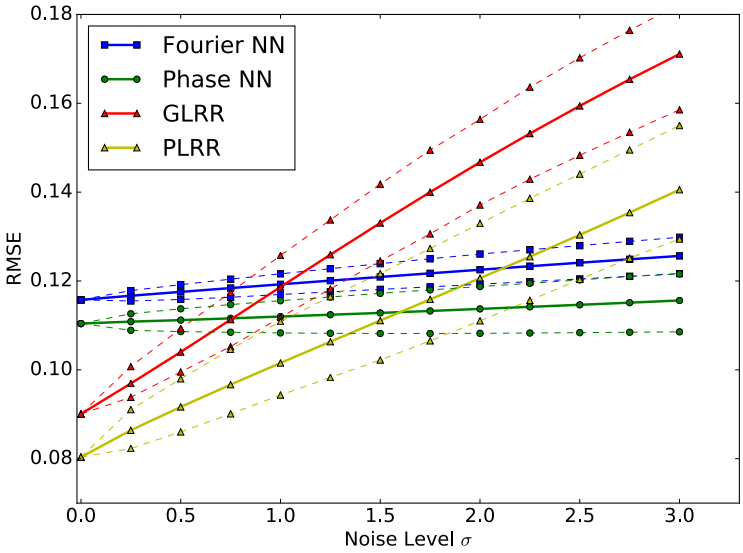

Figure 4.7: RMSE on the Aerosol test set, corrupted by various levels of noise averaged over 100 runs, with the 5^{th} and the 95^{th} percentile. The noiseless case is shown with one run. RMSE from mean is 0.206.

AOD sensor measurements and each sample $\mathbf{x}_i^a$ is 16-dimensional. This can be understood as a distribution regression problem (overview in Section 1.2.1.4 of Chapter 1) where each bag is treated as a set of samples from some distribution.

We use 640 bags for training and 160 bags for testing. Here in the bags for testing *only*, we add varying levels of Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{Z})$ to each bag, where $\mathbf{Z}$ is a diagonal matrix with diagonal components $z_k \sim U[0, \sigma v_k]$ with v_k being the empirical variance in dimension k across all samples, accounting for different scales across dimensions. For comparisons, we consider linear ridge regression on embeddings with respect to a Gaussian kernel, approximated with RFF (GLRR) (i.e. a linear kernel is applied on approximate embeddings), linear ridge regression on phase features (PLRR) (i.e. normalisation step is applied to obtain (4.5)), and also the phase and Fourier neural networks (NN), described in Appendix 4.7.3, tuning all hyperparameters with 3-fold cross validation. With the same model, we now measure Root Mean Square Error (RMSE) 100 times with various noise-corrupted test sets and results are shown in Figure 4.7. It is also noted that a second level non-linear kernel $\tilde{K}$ does not improve performance significantly on this problem [Szabó et al., 2016].

We see that GLRR and PLRR are competitive (see Table 4.2) in the noiseless case, and these clearly outperform both the Fourier NN and Phase NN (likely due to the small size of the

Table 4.2: Average RMSE for the Aerosol Dataset across 10 runs, for different train and test splits, with standard deviation in brackets

	FOURIER NN	PHASE NN	GLRR	PLRR
NO NOISE	0.101 (0.011)	0.101 (0.008)	0.079 (0.010)	0.085 (0.009)

dataset). For increasing noise, the performance of GLRR degrades significantly, and while the performance of PLRR degrades also, the model is much more robust under additional SPD noise. In comparison, the Phase NN implementation is almost insensitive to covariate shift in the test sets, unlike the performance of PLRR, highlighting the importance of learning discriminative frequencies $\boldsymbol{\omega}$ in a very low signal-to-noise setting. It is noted that the Fourier NN performs similarly to that of the Phase NN on this example. Interestingly, discriminative frequencies $\boldsymbol{\omega}$ learnt on the training data correspond to Fourier features that are nearly normalised (i.e. they are close to unit norm - see Figure 4.8). This means that the Fourier NN has *learned to be approximately invariant* based on training data, indicating that the original Aerosol data potentially has irrelevant SPD noise components. This is reinforced by the nature of the dataset (each bag contains 100 randomly selected potentially cloudy pixels, known to be noisy [Wang et al., 2012]) and no loss of performance from going from GLRR to PLRR. The results highlights that phase features are stable under additive SPD noise.

4.5.2.3 Dark matter dataset

We now study the use of phase features on the dark matter dataset, composing of a catalog of galaxy clusters. In this setting, we would like to predict the total mass of galaxy clusters, using the dispersion of velocities in the direction along our line of sight. In particular, we will use the 'ML1' dataset, as obtained from the authors of Ntampaka et al. [2015, 2016], who constructed a catalog of massive halos from the MultiDark `mdpl` simulation [Klypin et al., 2014]. The dataset contains 5028 bags, with each sample consisting of its sub-object velocity and its mass label in $\mathbb{R}$. By viewing each galaxy cluster at multiple lines of sights, we obtain 15 000 bags. For experiments, we use approximately 9000 bags for training, and 3000 bags each for validation and testing, keeping those of multiple lines of sight in the same set. As before, we use GLRR and PLRR and we also include in comparisons methods

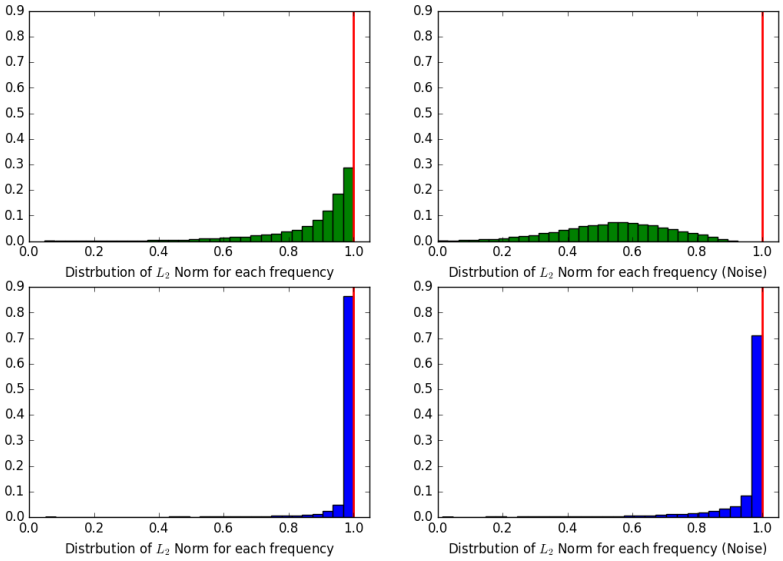

Figure 4.8: Histograms for the distribution of the L_2 norm of the averages of Fourier features over each frequency $\boldsymbol{\omega}$ for the original aerosol test set and the aerosol test set with added noise ($\sigma = 3$), here red line denotes the unit norm representing the phase features. **Top Green:** Random Fourier Features $\boldsymbol{\omega}$ (with the optimised kernel bandwidth). **Bottom Blue:** Learnt Fourier features $\boldsymbol{\omega}$ from the Fourier Neural Network.

with a second level Gaussian kernel (with RFF) applied to phase features (PGRR) and to approximate embeddings (GGRR). For a baseline, we also include a first level linear kernel (equivalent to representing each bag with its mean), before applying a second level Gaussian kernel (LGRR). We use the same set of randomly sampled frequencies across the methods, tuning for the scale of the frequencies and for regularisation parameters.

Table 4.3 shows the results of the methods across 10 different data splits, with 50 sets of randomised frequencies for each data split. We see that PLRR is significantly better than GLRR. This suggests that under this model structure, by removing SPD components from each bag, we can target the underlying signal and obtain superior performance, highlighting the applicability of phase features. Considering a second level Gaussian kernel, we see that the GGRR has a slight advantage over PGRR, with PGRR performing similar to PLRR. This suggests that the SPD components of the distribution of sub-object velocity may be useful for predicting the mass of a galaxy cluster if an additional nonlinearity is applied to embeddings – whereas the benefits of removing them outweigh the signal present in them without this additional nonlinearity. To show that indeed the phase features are robust to SPD components, we perform the same covariate shift experiment as in the aerosol dataset, with

Table 4.3: Mean Square Error (MSE) on dark matter dataset for 500 runs with 5^{th} and 95^{th} percentile.

Algorithm	MSE
Mean	0.16
PLRR	$\mathbf{0.021}$ $(0.018, 0.024)$
GLRR	0.033 $(0.030, 0.037)$
LGRR	0.032 $(0.028, 0.036)$
PGRR	0.021 $(0.017, 0.024)$
GGRR	$\mathbf{0.018}$ $(0.015, 0.019)$

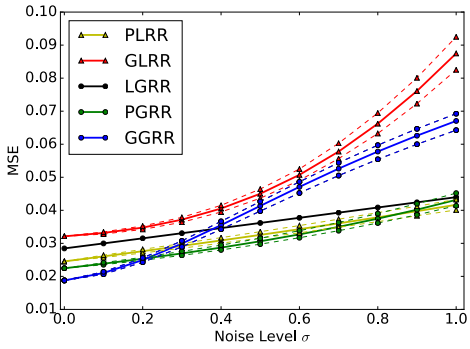

Figure 4.9: MSE with various levels of noise added on test set, with 5^{th} and 95^{th} percentile.

results given in Figure 4.9. Note that LGRR is robust to noise, as each bag is represented by its mean.

4.6 Conclusion

No dataset is immune from measurement noise and often this noise differs across different data generation and collection processes. When measuring distances between distributions, can we disentangle the differences in noise from the differences in the signal? We considered two different ways to encode invariances to additive symmetric noise in those distances, each with different strengths: a nonparametric measure of asymmetry in paired sample differences and a weighted distance between the empirical phase functions. The former was used to construct a hypothesis test on whether the difference between the two generating processes can be explained away by the difference in postulated noise, whereas the latter allowed us to introduce a flexible framework for invariant feature construction and learning algorithms on distribution inputs which are robust to measurement noise and target underlying signal distributions.

4.7 Chapter appendix

4.7.1 Phase Discrepancy and asymmetry in paired differences proofs

In this section, we will provide further details of the definitions, calculations and proofs in Section 4.3 and 4.4. Phase discrepancy is defined as the weighted L_2-distances between the phase functions, i.e.

$$\text{PhD}(X, Y) = \int \left|\rho_X(\boldsymbol{\omega}) - \rho_Y(\boldsymbol{\omega})\right|^2 d\Lambda(\boldsymbol{\omega}),$$

for some positive measure Λ (w.l.o.g. a probability measure). Phase discrepancy measures how much X and Y differ up to an independent SPD noise component.

We note that while the form of the PhD is motivated by that of the MMD (weighted L_2-distances between the characteristic functions), relating it to the properties of the corresponding kernel and its RKHS is not straightforward. For example, constructing a PhD interpretation as a supremum over the RKHS unit ball (which is often how MMD is introduced) is immediate only for the case where indecomposable parts are point masses. Namely, if $X = X_0 + U$ and $Y = Y_0 + V$, i.e. indecomposable parts are almost surely constant vectors x_0 and y_0, then

$$\text{PhD}(X, Y) = \|k(\cdot, \mathbf{x}_0) - k(\cdot, \mathbf{y}_0)\|^2_{\mathcal{H}_k} = \sup_{\|f\|_{\mathcal{H}_k} \leq 1} \left|f(\mathbf{x}_0) - f(\mathbf{y}_0)\right|^2,$$

by virtue of $\rho_X(\boldsymbol{\omega}) = e^{i\boldsymbol{\omega}^\top \mathbf{x}_0} = \varphi_{\mathbf{x}_0}(\boldsymbol{\omega})$. In other cases, while it is clear that the spectral properties of the kernel still regulate the amount of frequency content that is used, one obtains the RKHS distance between the kernel convolutions of the inverse Fourier transforms of the phase functions so the interpretation is less clear.

Below, we provide the proofs of the propositions from the main text.

Proposition 4.

$$\textit{PhD}(X, Y) = 2 - 2 \int \frac{\mathbb{E}[\cos\left(\boldsymbol{\omega}^\top (X - Y)\right)]}{\sqrt{\mathbb{E}[\cos\left(\boldsymbol{\omega}^\top (X - X')\right)]\mathbb{E}[\cos\left(\boldsymbol{\omega}^\top (Y - Y')\right)]}} d\Lambda(\boldsymbol{\omega}).$$

Proof.

$$\begin{aligned}
\mathrm{PhD}(X,Y) &= \int \left|\rho_X(\boldsymbol{\omega}) - \rho_Y(\boldsymbol{\omega})\right|^2 d\Lambda(\boldsymbol{\omega}) \\
&= \int \left|\rho_X(\boldsymbol{\omega})\right|^2 d\Lambda(\boldsymbol{\omega}) + \int \left|\rho_Y(\boldsymbol{\omega})\right|^2 d\Lambda(\boldsymbol{\omega}) - \int \left(\rho_X\overline{\rho_Y} + \overline{\rho_X}\rho_Y\right) d\Lambda(\boldsymbol{\omega}) \\
&= 2 - \int \frac{\varphi_X\overline{\varphi_Y} + \overline{\varphi_X}\varphi_Y}{|\varphi_X|\,|\varphi_Y|} d\Lambda(\boldsymbol{\omega}) \\
&= 2 - 2\int \frac{\varphi_Z}{\sqrt{\varphi_{X-X'}\varphi_{Y-Y'}}} d\Lambda(\boldsymbol{\omega}),
\end{aligned}$$

where X and X' are iid, Y and Y' are iid and Z is an equal mixture of $X-Y$ and $Y-X$. Indeed,

$$\varphi_X\overline{\varphi_Y} + \overline{\varphi_X}\varphi_Y = \varphi_{X-Y} + \varphi_{Y-X} = 2\varphi_Z,$$

and

$$\varphi_{X-X'} = \varphi_X\overline{\varphi_X} = |\varphi_X|^2.$$

Note that $X-X'$,$Y-Y'$ and Z are all symmetric. Thus,

$$\begin{aligned}
\varphi_Z(\boldsymbol{\omega}) &= \mathbb{E}\left[\cos\left(\boldsymbol{\omega}^\top Z\right)\right] = \frac{1}{2}\mathbb{E}\left[\cos\left(\boldsymbol{\omega}^\top (X-Y)\right)\right] + \frac{1}{2}\mathbb{E}\left[\cos\left(\boldsymbol{\omega}^\top (Y-X)\right)\right] \\
&= \mathbb{E}\left[\cos\left(\boldsymbol{\omega}^\top (X-Y)\right)\right].
\end{aligned}$$

Substituting provides us the result. □

Proposition 5. *$K_{\boldsymbol{\omega}}(P_X,P_Y) = \left(\frac{\mathbb{E}\xi_{\boldsymbol{\omega}}(X)}{\|\mathbb{E}\xi_{\boldsymbol{\omega}}(X)\|}\right)^\top \left(\frac{\mathbb{E}\xi_{\boldsymbol{\omega}}(Y)}{\|\mathbb{E}\xi_{\boldsymbol{\omega}}(Y)\|}\right)$ is a positive definite kernel on probability measures $\forall\boldsymbol{\omega}$, where here $\xi_{\boldsymbol{\omega}}(x) = \left[\cos\left(\boldsymbol{\omega}^\top x\right), \sin\left(\boldsymbol{\omega}^\top x\right)\right]$, so is $K(P_X,P_Y) = \int K_{\boldsymbol{\omega}}(P_X,P_Y)\, d\Lambda(\boldsymbol{\omega})$ for any positive measure Λ.*

Proof. Define a feature map $\xi_{\boldsymbol{\omega}} : \mathcal{X} \to \mathbb{R}^2$ with $\xi_{\boldsymbol{\omega}}(x) = \left[\cos\left(\boldsymbol{\omega}^\top x\right), \sin\left(\boldsymbol{\omega}^\top x\right)\right]$, which induces a kernel on $\mathcal{X}$ given by $k_{\boldsymbol{\omega}}(x,y) = \cos\left(\boldsymbol{\omega}^\top (x-y)\right)$. Then, we have that

$$\kappa_{\boldsymbol{\omega}}(P_X,P_Y) = \mathbb{E}[\cos\left(\boldsymbol{\omega}^\top (X-Y)\right)] = \mathbb{E}[k_{\boldsymbol{\omega}}(X,Y)] = \left(\mathbb{E}\xi_{\boldsymbol{\omega}}(X)\right)^\top \mathbb{E}\xi_{\boldsymbol{\omega}}(Y)$$

is a valid kernel on probability measures and so is the normalised kernel

$$K_{\boldsymbol{\omega}}(P_X,P_Y) = \frac{\kappa_{\boldsymbol{\omega}}(P_X,P_Y)}{\sqrt{\kappa_{\boldsymbol{\omega}}(P_X,P_X)\,\kappa_{\boldsymbol{\omega}}(P_Y,P_Y)}} = \left(\frac{\mathbb{E}\xi_{\boldsymbol{\omega}}(X)}{\|\mathbb{E}\xi_{\boldsymbol{\omega}}(X)\|}\right)^\top \left(\frac{\mathbb{E}\xi_{\boldsymbol{\omega}}(Y)}{\|\mathbb{E}\xi_{\boldsymbol{\omega}}(Y)\|}\right),$$

where we used that $\mathbb{E}[\cos\left(\boldsymbol{\omega}^\top (X-X')\right)] = \left(\mathbb{E}\xi_{\boldsymbol{\omega}}(X)\right)^\top \mathbb{E}\xi_{\boldsymbol{\omega}}(X') = \|\mathbb{E}\xi_{\boldsymbol{\omega}}(X)\|^2$. For the last claim, simply note that integrating through the positive measure preserves positive semidefinitess, i.e. $\sum \alpha_i\alpha_j K(P_i,P_j) = \int \left(\sum \alpha_i\alpha_j K_{\boldsymbol{\omega}}(P_i,P_j)\right) d\Lambda(\boldsymbol{\omega}) \geq 0$. □

As a direct corollary,

Proposition 6. $\text{PhD}(X,Y) = 2 - 2K\left(P_X, P_Y\right) = 2\int\left(1 - \left(\frac{\mathbb{E}\xi_{\boldsymbol{\omega}}(X)}{\|\mathbb{E}\xi_{\boldsymbol{\omega}}(X)\|}\right)^{\top}\left(\frac{\mathbb{E}\xi_{\boldsymbol{\omega}}(Y)}{\|\mathbb{E}\xi_{\boldsymbol{\omega}}(Y)\|}\right)\right)d\Lambda(\boldsymbol{\omega})$.

Proposition 7. *Under the null hypothesis, $X - Y$ is SPD* $\iff$ $X_0 \overset{d}{=} Y_0$.

Proof. Under H_0, since X_0 has the same distribution as Y_0, then so do $X - Y = X_0 - Y_0 + U - V$ and $Y - X = Y_0 - X_0 + V - U$ as $U - V$ is symmetric. Moreover, $\varphi_{X-Y} = |\varphi_{X_0}|^2\varphi_U\varphi_V > 0$, so $X - Y$ is SPD. Conversely, if we assume that $X - Y$ is SPD, i.e. $\varphi_X\overline{\varphi_Y} > 0$, then $\rho_{X_0}\overline{\rho_{Y_0}} > 0$. Since $|\rho_{X_0}| = |\rho_{Y_0}| = 1$, this implies that $\rho_{X_0} = \rho_{Y_0}$, and hence $X_0 \overset{d}{=} Y_0$, since we assumed that X_0 and Y_0 belong to $\mathcal{P}(\mathbb{R}^d)$. Hence, we have that $X - Y$ is SPD $\iff$ $X_0 \overset{d}{=} Y_0$. □

4.7.2 Paired differences

Another way to measure asymmetry of the difference between random vectors X and Y is to use $\text{MMD}(X - Y, Y - X)$ instead of $\text{PhD}(X, Y)$. However, this quantity is not invariant, i.e., $\text{MMD}(X - Y, Y - X) \neq \text{MMD}(X_0 - Y_0, Y_0 - X_0)$, and in fact the values will heavily depend on the distributions of U and V. We note that

$$\varphi_{X-Y}(\boldsymbol{\omega}) - \varphi_{Y-X}(\boldsymbol{\omega}) \;=\; 2i\mathbb{E}[\sin\left(\boldsymbol{\omega}^{\top}\left(X - Y\right)\right)],$$

so that we are effectively measuring the size of the imaginary part of the characteristic function of $X - Y$ (which should not be there if it is symmetric). There are several different ways in which we can write this quantity:

$$\begin{aligned}
\text{MMD}(X - Y, Y - X) &= \|\mathbb{E}[k(\cdot, X - Y)] - \mathbb{E}[k(\cdot, Y - X)]\|^2_{\mathcal{H}_k} \\
&= \int |\varphi_X\overline{\varphi_Y} - \overline{\varphi_X}\varphi_Y|^2 \, d\Lambda(\boldsymbol{\omega}) \\
&= 4\int \mathbb{E}[\sin\left(\boldsymbol{\omega}^{\top}\left(X - Y\right)\right)]^2 d\Lambda(\boldsymbol{\omega}) \\
&= \int |\varphi_X|^2 |\varphi_Y|^2 \left(2 - \frac{\varphi_X\overline{\varphi_Y}}{\overline{\varphi_X}\varphi_Y} - \frac{\overline{\varphi_X}\varphi_Y}{\varphi_X\overline{\varphi_Y}}\right) d\Lambda(\boldsymbol{\omega}).
\end{aligned}$$

The last expression indicates that this quantity is affected by the amplitude of the individual characteristic functions, with experimental details to show this in Section 4.5.2.1. Moreover, the quantity does not appear to lend itself to the *feature on distributions* formalism, i.e. we were unable to derive some Hilbert space features $\Upsilon(P) \in \mathcal{H}$ such that $\text{MMD}(X - Y, Y -$

$X) = \|\Upsilon(P_X) - \Upsilon(P_Y)\|^2_{\mathcal{H}}$, and it is thus unclear whether this approach can be used to define a valid kernel on distributions.

4.7.3 Learning discriminative features

Algorithm 1 Phase/Fourier Neural Network

Input: Batch of bag of samples $\mathbf{X} \in \mathbb{R}^{b \times N \times d}$, where b is the batch size, N is the bag size and d is the dimension
Output: Classification or Regression Output
1. Compute $f(\mathbf{X}) = \mathbf{XW}$ where $\mathbf{W} \in \mathbb{R}^{d \times J}$
2. Apply a $\sin$ and $\cos$ activation function, i.e. $l_1(X) = [\sin(f(\mathbf{X})) \cos(f(\mathbf{X}))]$
3. Apply mean pooling operation over N, effectively computing $\hat{\mathbb{E}}\xi_{\boldsymbol{\omega}_i}(\mathbf{X})$ for each $\boldsymbol{\omega}_i \in \mathbb{R}^d$

$$l_2(X) = \left[\hat{\mathbb{E}}\xi_{\boldsymbol{\omega}_1}(\mathbf{X}), \ldots, \hat{\mathbb{E}}\xi_{\boldsymbol{\omega}_J}(\mathbf{X})\right] \in \mathbb{R}^{2J}$$

4. For Phase Neural Network, compute $\left\|\hat{\mathbb{E}}\xi_{\boldsymbol{\omega}_1}(\mathbf{X})\right\|$ for each frequency and normalise to obtain:

$$l_3(X) = \left[\frac{\hat{\mathbb{E}}\xi_{\boldsymbol{\omega}_1}(\mathbf{X})}{\left\|\hat{\mathbb{E}}\xi_{\boldsymbol{\omega}_1}(\mathbf{X})\right\|}, \ldots, \frac{\hat{\mathbb{E}}\xi_{\boldsymbol{\omega}_J}(\mathbf{X})}{\left\|\hat{\mathbb{E}}\xi_{\boldsymbol{\omega}_J}(\mathbf{X})\right\|}\right]$$

5. Batch Normalisation Layer
6. Output layer

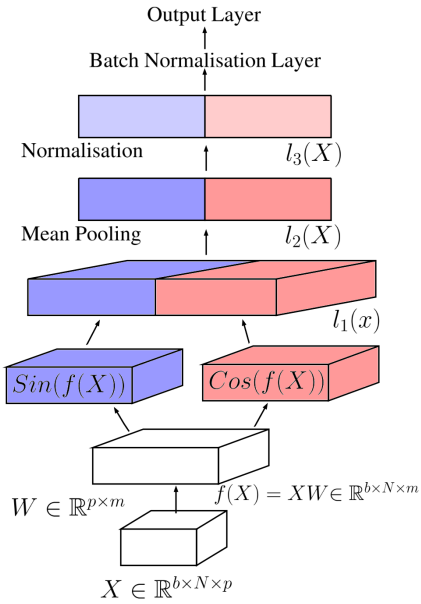

Figure 4.10: Main structure of the phase neural network.

Algorithm 1 shows the phase Neural Network (phase NN) and the Fourier Neural Network (Fourier NN), where the latter can be obtained by simply removing step 4 in the algorithm. Although the batch normalisation is not required, it is highly recommended for faster training of the network [Ioffe & Szegedy, 2015], due to the normalisation for the phase neural network in step 5 of the algorithm. Because of the neural network structure, we can take advantage of the rich literature, as well as alter the network in order to target a variety of different problems. For example, setting now the loss function as the squared loss, cross entropy or pinball loss, we can solve tasks in regression, classification or quantile regression on distributional inputs with discriminative frequencies.

4.7.4 Characteristic and Phase function plots

The red points denote the empirical characteristic/phase function constructed with 750 frequencies from a Gaussian kernel with $\sigma = 2$ using a bag size of 1000 observations, with some additional Gaussian noise.

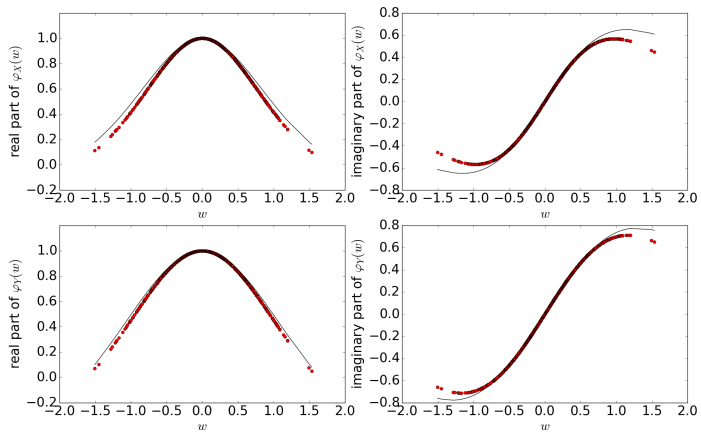

Figure 4.11: The black line here correspond to the real and imaginary part of the true characteristic function of the $\chi^2(4)/4$ and $\chi^2(8)/8$ distribution, denoted X, Y on the top and bottom graphs respectively.

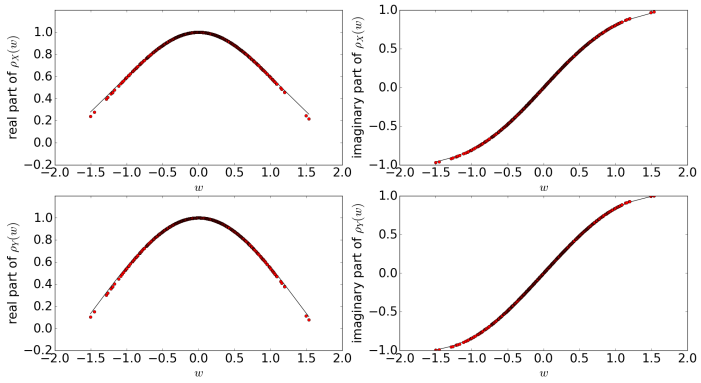

Figure 4.12: The black line here correspond to the real and imaginary part of the true phase function of the $\chi^2(4)/4$ and $\chi^2(8)/8$ distribution, denoted X, Y on the top and bottom graphs respectively.

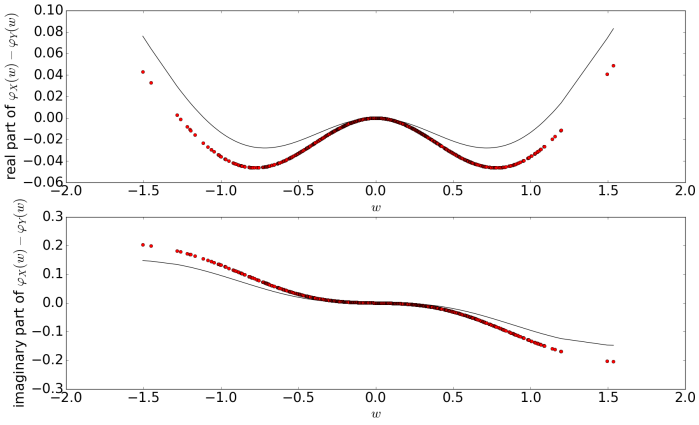

Figure 4.13: The top and bottom graph denotes the difference in the real and imaginary part of the characteristic function for the $\chi^2(4)/4$ and $\chi^2(8)/8$ as in Figure 4.11.

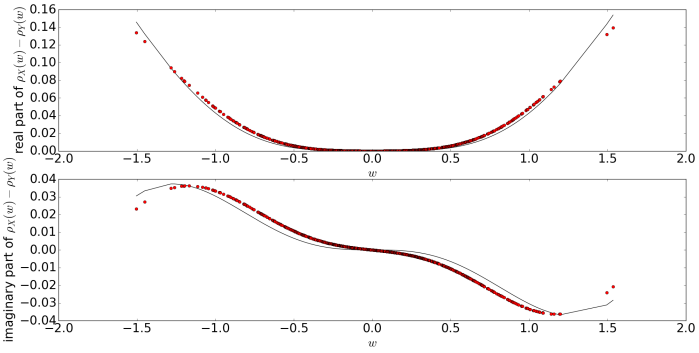

Figure 4.14: The top and bottom graph denotes the difference in the real and imaginary part of the phase function for the $\chi^2(4)/4$ and $\chi^2(8)/8$ as in Figure 4.12.

4.7.5 Implementation details

PhD two-sample test For the PhD two-sample test for the toy dataset, for each of the 1000 runs, we use a permutation size of 400, with the number of frequencies sampled set at 50. Here the frequencies are sampled using the radial frequency distribution, where $\mathbf{\Sigma}$ is chosen to be $\sigma^2\mathbf{I}$, with σ^2 being the empirical variance of the two set of samples. The Radial Frequency Distribution is defined as follows:

$$\mathbf{w} = \mathbf{R}\mathbf{\Sigma}^{-\frac{1}{2}}\boldsymbol{\psi}$$

where $\boldsymbol{\psi} \in \mathbb{R}^n$ is uniformly distributed on the L_2 unit sphere $\mathcal{S}_{n-1}$, and $R \in \mathbb{R}_+$ is a radius drawn independently from a folded Gaussian $\mathcal{N}^+(0,1)$. The radial frequency distribution is useful in high dimensions, as unlike the normal distributions, which 'under samples' the low or middle frequencies, it is able to sample a broader range of frequencies due to its form. By

covering a broader range of frequencies, we may be able to 'better encode' information of the distribution represented by the bags, leading to a feature map that is more informative.

Aerosol dataset For the network, we use a squared loss function with an additional L_2 weight decay for regularisation, with a separate regularisation parameter for the two individual layers. For optimisation, we again use ADAM [Kingma & Ba, 2015] with fixed learning rate decay and 120 epochs, with a batch size of 10. We perform a 3-fold cross validation, and compute the MSE. We tune the learning rate, regularisation parameters and also number of frequencies for the neural network, here we initialise the first layer with Gaussian distribution with standard deviation = $1/\gamma_0$, where γ_0 denote the median heuristic. Regarding the experiment for the extraction of frequencies due the similar behaviour of Fourier and Phase NN, we extract the frequencies $\boldsymbol{\omega}$ learnt and compute $\left\|\hat{\mathbb{E}}\xi_{\boldsymbol{\omega}}(\mathbf{X})\right\|$ for each frequency over the original and noisy test set, similarly we do this for the frequencies generated from the Gaussian kernel (with the optimised bandwidth on the original aerosol dataset).

Dark Matter dataset For all methods we sample frequencies from the normal distribution (with standard deviation = $1/\gamma_0$, where γ_0 denote the median heuristic.). After sampling a set of frequencies, we tune the scale of the set of frequencies and also the ridge regularisation parameter using the validation set. In particular we use 75 frequencies on the first and second level of the kernel whenever they are used. Note we use the same set of frequencies (at each individual kernel level) across all the methods in a single run to allow for easier comparison, with potentially different scale tuned on the validation set.

Chapter 5

A Differentially Private Kernel Two-Sample Test

This chapter is based on the following paper below, here we *exclude* the proofs to be found in the Appendix of the following paper, as they are not my original contribution.

Anant, Raj*, Ho Chung Leon Law*, Dino Sejdinovic, and Mijung Park
A Differentially Private Kernel Two-Sample Test [Raj et al., 2019]
European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), 2019

Kernel two-sample testing is a useful statistical tool in determining whether data samples arise from different distributions without imposing any parametric assumptions on those distributions. However, raw data samples can expose sensitive information about individuals who participate in scientific studies, which makes the current tests vulnerable to privacy breaches. Hence, we design a new framework for kernel two-sample testing conforming to differential privacy constraints, in order to guarantee the privacy of subjects in the data. Unlike existing differentially private parametric tests that simply add noise to data, kernel based testing imposes a challenge due to a complex dependence of test statistics on the raw data, as these statistics correspond to estimators of distances between representations of probability measures in Hilbert spaces. Our approach considers finite dimensional approximations to those representations. As a result, a simple χ^2 test is obtained, where the test statistic depends on the mean and covariance of empirical differences between the samples, which we

* denote authors with equal contribution.

perturb for a privacy guarantee. We investigate the utility of our framework in two realistic settings and conclude that our method requires only a relatively modest increase in sample size to achieve a similar level of power to the non-private tests in both settings.

5.1 Introduction

Several recent works suggest that it is possible to identify subjects that have participated in scientific studies based on publicly available aggregate statistics (cf. Homer et al. [2008] and Johnson & Shmatikov [2013] among many others). The *differential privacy* formalism [Dwork et al., 2006b] provides a way to quantify the amount of information on whether or not a single individual's data is included (or modified) in the data and also provides rigorous privacy guarantees in the presence of *arbitrary side information*.

An important tool in statistical inference is *two-sample testing*, in which samples from two probability distributions are compared in order to test the null hypothesis that the two underlying distributions are identical against the general alternative that they are different. In this chapter, we focus on the nonparametric, *kernel based* two-sample testing approach and investigate the utility of this framework in a differentially private setting. The kernel based two-sample testing was introduced by Gretton et al. [2007, 2012a] who considers an estimator of maximum mean discrepancy (MMD) [Borgwardt et al., 2006], the distance between embeddings of probability measures in a reproducing kernel Hilbert space (RKHS) (see Muandet et al. [2017] for a recent review), as a test statistic for the nonparametric two-sample problem.

Many existing differentially private testing methods are based on categorical data, i.e. counts [Gaboardi et al., 2016; Gaboardi & Rogers, 2017; Rogers & Kifer, 2017], in which case a natural way to achieve privacy is to simply add noise to these counts. However, when we consider a more general input space $\mathcal{X}$ for testing, the amount of noise needed to privatise the data essentially becomes the order of diameter of the input space (explained in Appendix 5.8.1). For spaces such as $\mathbb{R}^d$, the level of noise needed can destroy the utility of the data as well as that of the test (hence approaches such as in Chapter 4 is not feasible).

Here we take an alternative approach and privatise only the quantities that are required for the test. In particular, for the kernel two-sample testing, we only require the empirical kernel embedding $\frac{1}{N}\sum_i k(\mathbf{x}_i, \cdot)$ corresponding to a dataset, where $\mathbf{x}_i \in \mathcal{X}$ and k is some positive definite kernel. Now, as the kernel embedding lives in $\mathcal{H}_k$, a space of functions, a natural way to protect them is to add Gaussian process noise as suggested in Hall et al. [2013]. Although sufficient for situations where the functions themselves are of interest, embeddings impaired by a Gaussian process do not lie in the same RKHS [Wahba, 1990], and hence one cannot estimate the RKHS distances between such noisy embeddings. Alternatively, one could consider adding noise to an estimator of MMD [Gretton et al., 2012a]. However, asymptotic null distributions of these estimators are data dependent and the test thresholds are typically computed by permutation testing or by eigendecomposing centred kernel matrices of the data [Gretton et al., 2009]. In this case neither of these approaches is available in a differentially private setting as they both require further access to data.

Contribution In this chapter, we build a differentially private kernel two-sample testing framework, by considering *analytic representations* of probability measures [Chwialkowski et al., 2015; Jitkrittum et al., 2016], aimed at large scale testing scenarios. Through this formulation, we are able to obtain a test statistic that is based on means and covariances of feature vectors of the data. This suggests that privatisation of summary statistics of the data is sufficient to make the tests differentially private, implying a reduction of level of noise needed versus adding to the data directly (as summary statistics are less sensitive to individual changes). Further, we show that while the asymptotic distribution under the null hypothesis of the test statistic does not depend on the data, unlike the non-private case, using the asymptotic null distribution can lead to grossly miscalibrated Type I control. Hence, we propose a remedy for this problem, and give approximations of the finite-sample null distributions, yielding good Type I error control and power-privacy tradeoffs experimentally in Section 5.6.

5.1.1 Related work

To the best of our knowledge, this work is the *first* to propose a kernel two-sample test in a differential private setting. Although, there are various differentially private hypothesis

test in the literature, most of these revolve around categorical data [Gaboardi et al., 2016; Gaboardi & Rogers, 2017; Rogers & Kifer, 2017] on χ^2 tests. This is very different to our work, which considers a null hypothesis of equal distributions against a general alternative hypothesis. Further, while there are several works that connect kernel methods with differential privacy, including Jain & Thakurta [2013], Hall et al. [2013] and Balog et al. [2017], none of these attempt to make the kernel based two-sample testing procedure private. It is also important to emphasise that in a hypothesis testing, it is not sufficient to make the test statistic differentially private, as one has to carefully construct the distribution under the null hypothesis in a differential private manner, taking into account the level of noise added.

5.1.2 Motivation and setting

We now present the two privacy scenarios that we consider and motivate their usage. In the first scenario, we assume there is a trusted curator and also an untrusted tester, which we want to protect data from. In this setting, the trusted curator has access to the two datasets and computes the mean and covariance of the empirical differences between the feature vectors. The curator can protect the data in two different ways: (1) perturb mean and covariance separately and release them; or (2) compute the statistic without perturbations and add noise to it directly. The tester can now take these perturbed quantities and performs the test at a desired significance level. Here, we separate the entities of tester and curator, as sometimes a decision whether to reject or not is of interest (which can be done by the trusted curator alone), for example one can imagine that the tester may require the test-statistic/p-values for multiple hypothesis testing corrections. In the second scenario, we assume that there are two data-owners, each having one dataset each, and a tester. In this case, as no party trusts the other, each data-owner has to perturb their own mean and covariance of the feature vectors and release them to the tester. Under these two settings, we will exploit various differentially private mechanisms and empirically study the utility of the proposed framework.

5.2 Background

First introduced by Chwialkowski et al. [2015] and then extended and further analysed by Jitkrittum et al. [2016], here we will focus on kernel two-sample test based on the mean embedding (ME) and on an approach based on the smooth characteristic function (SCF). The corresponding notation and background can be found in Section 1.2.1.3 in Chapter 1. Throughout this chapter, we will assume that we use bounded and nonnegative kernels in the ME test (e.g. Gaussian and ARD Kernel), in particular $0 \leq k(\mathbf{x}, \mathbf{y}) \leq \kappa, \ \ \forall \mathbf{x}, \mathbf{y}$, and that the weighting function in the SCF test is also bounded: $0 \leq g(\mathbf{x}) \leq \kappa/2$ (note the additional division). This implies that in both cases, we have $||\mathbf{z}_i||_2 \leq \kappa\sqrt{J}$, for any $i = 1, \ldots, N$.

5.2.1 Differential privacy

Given an algorithm $\mathcal{M}$ and neighbouring datasets $\mathcal{D}$, $\mathcal{D}'$ differing by a single data sample, the *privacy loss* of an outcome o is

$$L^{(o)} = \log\left(\frac{Pr(\mathcal{M}_{(\mathcal{D})} = o)}{Pr(\mathcal{M}_{(\mathcal{D}')} = o)}\right). \tag{5.1}$$

The mechanism $\mathcal{M}$ is called ϵ-DP if and only if $|L^{(o)}| \leq \epsilon, \forall o, \mathcal{D}, \mathcal{D}'$. A weaker version of the above is (ϵ, δ)-DP, if and only if $|L^{(o)}| \leq \epsilon$, with probability at least $1 - \delta$. The definition states that a single individual's participation in the data do not change the output probabilities by much, and hence this limits the amount of information that the algorithm reveals about any one individual.

A differentially private algorithm is designed by adding noise to the algorithms' outputs. Suppose a deterministic function $h : \mathcal{D} \mapsto \mathbb{R}^p$ computed on sensitive data $\mathcal{D}$ outputs a p-dimensional vector quantity. In order to make h private, we can add noise in function h, where the level of noise is calibrated to the *global sensitivity* GS_h [Dwork et al., 2006a], defined by the maximum difference in terms of L_2-norm $||h(\mathcal{D}) - h(\mathcal{D}')||_2$, for neighbouring $\mathcal{D}$ and $\mathcal{D}'$ (i.e. differ by one data sample). In the case of Gaussian mechanism (Theorem 3.22 in Dwork & Roth [2014]), the output is perturbed by

$$\tilde{h}(\mathcal{D}) = h(\mathcal{D}) + \mathcal{N}(0, GS_h^2\sigma^2\mathbf{I}_p).$$

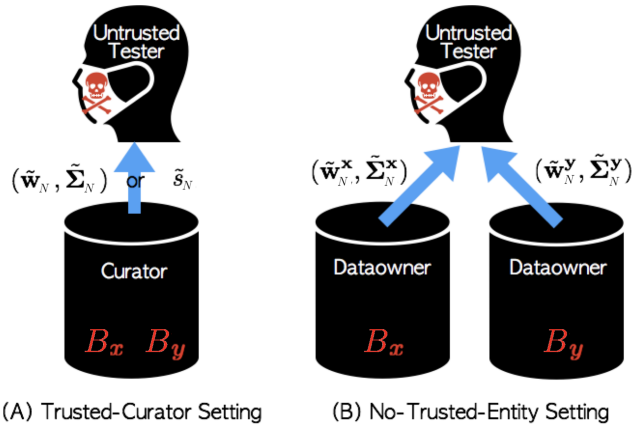

Figure 5.1: Two privacy settings. **(A)** A trusted curator releases a private test statistic or private mean and covariance of empirical differences between the features. **(B)** Data owners release private feature means and covariances calculated from their samples. In both cases, an untrusted tester performs a test using the private quantities.

The perturbed function $\tilde{h}(\mathcal{D})$ is then (ϵ, δ)-DP, when $\sigma \geq \sqrt{2\log(1.25/\delta)}/\epsilon$, for $\epsilon \in (0, 1)$. When constructing our tests, we will use two important properties of differential privacy. The composability theorem [Dwork et al., 2006a] tells us that the strength of privacy guarantee degrades with the repeated use of DP-algorithms. In particular, when two differentially private subroutines are combined, where each one guarantees (ϵ_1, δ_1)-DP and (ϵ_2, δ_2)-DP respectively by adding independent noise, the parameters are simply composed by $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$. Furthermore, post-processing invariance [Dwork et al., 2006a] tells us that the composition of any arbitrary data-independent mapping with an (ϵ, δ)-DP algorithm is also (ϵ, δ)-DP. In the next section, we discuss the two privacy settings which we are considering for our study in this chapter.

5.2.2 Privacy settings

We now consider the two different privacy settings as shown in Figure 5.1.

(A) Trusted-curator (TC) setting

There is a trusted entity called curator that handles the datasets and outputs the private test statistic, either in terms of the perturbed $\tilde{\mathbf{w}}_N$ and $\tilde{\mathbf{\Sigma}}_N$, or in terms of the perturbed test statistic $\tilde{s}_N$. An untrusted tester performs a χ^2-test given these quantities.

(B) No-trusted-entity (NTE) setting

Each data owner outputs private mean and covariance of the feature vectors computed on

their own dataset, meaning that the owner of dataset B_x outputs $\tilde{\mathbf{w}}_N^{\mathbf{x}}$ and $\tilde{\mathbf{\Sigma}}_N^{\mathbf{x}}$ and the owner of dataset B_y outputs $\tilde{\mathbf{w}}_N^{\mathbf{y}}$ and $\tilde{\mathbf{\Sigma}}_N^{\mathbf{y}}$. An untrusted tester performs a χ^2 test given these quantities.

It is worth noting that the NTE setting is different from the typical two-party model considered in the differential privacy literature. In the two-party model, it is typically assumed that Alice owns a dataset B_x and Bob owns a dataset B_y, and they wish to compute some function $f(B_x, B_y)$ in a differentially private manner. In this case, the interest is to obtain a two-sided ϵ-differentially private protocol for f, i.e., each party's view of the protocol should be a differentially private function of the other party's input. For instance, the probability of Alice's views conditioned on B_y and $B_{y'}$ should be e^{ϵ} multiplicatively close to each other, where B_y and $B_{y'}$ are adjacent datasets [McGregor et al., 2010; Goyal et al., 2016]. On the other hand, in our NTE setting, we are interested in the case where each of the data owners releases DP statistics, where we would like to analyse how the performance of the test run by an untrusted third party using those DP statistics degrades with the level of DP in the released statistics.

5.3 Trusted-curator setting

In this setting, a trusted curator releases either a private test statistic or private mean and covariance, which then a tester can use to perform a χ^2 test. Given a total privacy budget (ϵ, δ), when we perturb mean and covariance separately, we spend (ϵ_1, δ_1) for mean perturbation and (ϵ_2, δ_2) for covariance perturbation, such that $\epsilon = \epsilon_1 + \epsilon_2$ and $\delta = \delta_1 + \delta_2$.

5.3.1 Perturbing mean and covariance

5.3.1.1 Mean perturbation

We obtain a private mean by adding Gaussian noise based on the analytic Gaussian mechanism recently proposed in Balle & Wang [2018]. The main reason for using this Gaussian mechanism over the original [Dwork & Roth, 2014] is that it provides a DP guarantee with smaller noise.

For $\mathbf{w}_N : \mathcal{D} \rightarrow \mathbb{R}^J$ that has the global L2-sensitivity $GS_2(\mathbf{w}_N)$, the analytic Gaussian mechanism produces $\tilde{\mathbf{w}}_N(\mathcal{D}) = \mathbf{w}_N(\mathcal{D}) + \mathbf{n}, \quad$ where $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma_N^2 \mathbf{I})$. Then $\tilde{\mathbf{w}}_N(\mathcal{D})$ is (ϵ_1, δ_1)-differentially private mean vector if σ_N follows the regime of Balle & Wang [2018]. Here implicitly σ_N depends on $GS_2(\mathbf{w}_N), \epsilon_1$ and δ_1. Assuming an entry difference between two parts of datasets $\mathcal{D} = (B_x, B_y)$ and $\mathcal{D}' = (B'_x, B'_y)$ (difference in *one* of B'_x or B'_y only) the global sensitivity is simply:

$$\begin{aligned} GS_2(\mathbf{w}_N) &= \max_{\mathcal{D}, \mathcal{D}'} \|\mathbf{w}_N(\mathcal{D}) - \mathbf{w}_N(\mathcal{D}')\|_2 \\ &= \max_{\mathbf{z}_i, \mathbf{z}'_i} \tfrac{1}{N} \|\mathbf{z}_i - \mathbf{z}'_i\|_2 \leq \tfrac{\kappa\sqrt{J}}{N}. \end{aligned} \tag{5.2}$$

where $\mathbf{z}_i$ is as the corresponding feature maps defined in Section 1.2.1.3 in Chapter 1.

5.3.1.2 Covariance perturbation

To obtain a private covariance matrix, we consider Dwork et al. [2014] which utilises Gaussian noise. Here since the covariance matrix is given by $\mathbf{\Sigma}_N = \mathbf{\Lambda}_N - \frac{N}{N-1} \mathbf{w}_N \mathbf{w}_N^\top$, where $\mathbf{\Lambda}_N = \frac{1}{N-1} \sum_{i=1}^N \mathbf{z}_i \mathbf{z}_i^\top$, we can simply privatise the covariance by simply perturbing the 2nd-moment matrix $\mathbf{\Lambda}_N$ and use the private mean $\tilde{\mathbf{w}}_N$, i.e., $\tilde{\mathbf{\Sigma}}_N = \tilde{\mathbf{\Lambda}}_N - \frac{N}{N-1} \tilde{\mathbf{w}}_N \tilde{\mathbf{w}}_N^\top$. To construct the 2nd-moment matrix $\tilde{\mathbf{\Lambda}}_N$ that is (ϵ_2, δ_2)-differentially private, we use $\tilde{\mathbf{\Lambda}}_N = \mathbf{\Lambda}_N + \mathbf{\Psi}$, where $\mathbf{\Psi}$ is obtained as follows:

1. Sample from $\boldsymbol{\eta} \sim \mathcal{N}(0, \beta^2 \mathbf{I}_{J(J+1)/2})$, where β is a function of $GS(\mathbf{\Lambda}_N), \epsilon_2, \delta_2$, outlined in the Appendix of Raj et al. [2019].
2. Construct an upper triangular matrix (including diagonal) with entries from $\boldsymbol{\eta}$.
3. Copy the upper part to the lower part so that resulting matrix $\mathbf{\Psi}$ becomes symmetric.
4. We perform an eigen-decomposition, and adjust negative eigenvalues to be $\epsilon > 0$, before reconstruction to ensure that Ψ is positive semi-definite.

The composability theorem [Dwork et al., 2006a] gives us $\tilde{\mathbf{\Sigma}}_N$ is (ϵ, δ)-differentially private.

5.3.2 Perturbing test statistic

The trusted-curator can also release a differentially private statistic, to do this we use the analytic Gaussian mechanism as before, perturbing the statistic by adding Gaussian noise. To use the mechanism, we need to calculate the global sensitivity needed of the test statistic $S_N = \mathbf{w}_N^\top(\mathbf{\Sigma}_N + \gamma_N \mathbf{I})^{-1}\mathbf{w}_N$, which we provide in this Theorem (proof to be found in the Appendix of Raj et al. [2019]):

Theorem 8. *Given the definitions of $\mathbf{w}_N$ and $\mathbf{\Lambda}_N$, and the L2-norm bound on $\mathbf{z}_i$'s, the global sensitivity $GS_2(S_N)$ of the test statistic S_N is $\frac{4\kappa^2 J\sqrt{J}}{N\gamma_N}\left(1 + \frac{\kappa^2 J}{N-1}\right)$, where γ_N is a regularisation parameter, which we set to be smaller than the smallest eigenvalue of $\mathbf{\Lambda}_N$.*

5.4 No-trusted-entity setting

In this setting, the two samples $\{\mathbf{x}_i\}_{i=1}^{N_x} \sim P_X$ and $\{\mathbf{y}_i\}_{i=1}^{N_y} \sim P_Y$ reside with different data owners, and here both owners wish to protect their samples in a differentially private manner. Note that in this context we allow the size of each sample to be different. The data owners first need to agree on the given kernel k as well as on the test locations $\{\mathbf{t}_j\}_{j=1}^{J}$. We denote now $\mathbf{z}_i^{\mathbf{x}} = \Big[k(\mathbf{x}_i, \mathbf{t}_1), \cdots, k(\mathbf{x}_i, \mathbf{t}_J)\Big]^\top$ in the case of the ME test or

$$\mathbf{z}_i^{\mathbf{x}} = \Big[g(\mathbf{x}_i)\cos(\mathbf{x}_i^\top \boldsymbol{\omega}_j), g(\mathbf{x}_i)\sin(\mathbf{x}_i^\top \boldsymbol{\omega}_j)\Big]_{j=1}^{J/2}$$

in the case of the SCF test (with $\{\boldsymbol{\omega}_j\}_{j=1}^{J/2}$ frequencies). Also, we denote

$$\mathbf{w}_{N_\mathbf{x}}^{\mathbf{x}} = \frac{1}{N_x}\sum_{i=1}^{N_x}\mathbf{z}_i^{\mathbf{x}} \qquad \mathbf{\Sigma}_{N_x}^{\mathbf{x}} = \frac{1}{N_x - 1}\sum_{i=1}^{N_x}(\mathbf{z}_i^{\mathbf{x}} - \mathbf{w}_{N_x}^{\mathbf{x}})(\mathbf{z}_i^{\mathbf{x}} - \mathbf{w}_{N_x}^{\mathbf{x}})^\top$$

and similarly for the sample $\{\mathbf{y}_i\}_{i=1}^{N_y} \sim P_Y$. The respective means and covariances $\mathbf{w}_{N_x}^{\mathbf{x}}$, $\mathbf{\Sigma}_{N_x}^{\mathbf{x}}$ and $\mathbf{w}_{N_y}^{\mathbf{y}}$, $\mathbf{\Sigma}_{N_y}^{\mathbf{y}}$ are computed by their data owners, which are then impaired independently with noise according to the sensitivity analysis described in Section 5.3.1. As a result we obtain differentially private means and covariances $\tilde{\mathbf{w}}_{N_x}^{\mathbf{x}}$, $\tilde{\mathbf{\Sigma}}_{N_x}^{\mathbf{x}}$ and $\tilde{\mathbf{w}}_{N_y}^{\mathbf{y}}$, $\tilde{\mathbf{\Sigma}}_{N_y}^{\mathbf{y}}$ at their respective users. All these quantities are then released to the tester whose role is to compute the test statistic and the corresponding p-value. In particular, the tester uses the statistic given by

$$\tilde{S}_{N_x,N_y} = \frac{N_x N_y}{N_x + N_y}(\tilde{\mathbf{w}}_{N_x}^{\mathbf{x}} - \tilde{\mathbf{w}}_{N_y}^{\mathbf{y}})^\top(\tilde{\mathbf{\Sigma}}_{N_x,N_y} + \gamma_N \mathbf{I})^{-1}(\tilde{\mathbf{w}}_{N_x}^{\mathbf{x}} - \tilde{\mathbf{w}}_{N_y}^{\mathbf{y}}),$$

where:

$$\tilde{\boldsymbol{\Sigma}}_{N_x,N_y} = \frac{(N_x-1)\tilde{\boldsymbol{\Sigma}}^{\mathbf{x}}_{N_x} + (N_y-1)\tilde{\boldsymbol{\Sigma}}^{\mathbf{y}}_{N_y}}{N_x+N_y-2}$$

is the pooled covariance estimate.

5.5 Analysis of null distributions

In the previous sections, we discussed the necessary tools to make the kernel two-sample tests private in two different settings by considering sensitivity analysis of quantities of interest. Here we consider the distributions of the statistics under $H_0 : P_X \stackrel{d}{=} P_Y$ for each of the two settings.

5.5.1 Trusted-curator setting: perturbed mean and covariance

In this scheme, noise is added both to the mean vector $\mathbf{w}_N$ and to the covariance matrix $\boldsymbol{\Sigma}_N$ (by dividing the privacy budget between these two quantities). Let us denote the perturbed mean by $\tilde{\mathbf{w}}_N$ and perturbed covariance with $\tilde{\boldsymbol{\Sigma}}_N$. The noisy version of the test statistic $\tilde{S}_N$ is given by:

$$\tilde{S}_N = N\tilde{\mathbf{w}}_N^\top \left(\tilde{\boldsymbol{\Sigma}}_N + \gamma_N \mathbf{I}\right)^{-1} \tilde{\mathbf{w}}_N \tag{5.3}$$

where γ_N is a regularisation parameter just like in the non-private statistic (1.15). We show below that the asymptotic null distribution (as sample size $N \to \infty$) of this private test statistic is in fact identical to that of the non-private test statistic. Intuitively, this is to be expected: as the number of samples increases, the contribution to the aggregate statistics of any individual observation diminishes, and the variance of the added noise goes to zero.

Theorem 9. *Assuming the Gaussian noise for $\tilde{\mathbf{w}}_N$ with the sensitivity bound in* (5.2) *and the perturbation mechanism introduced in Section 5.3.1 for $\tilde{\boldsymbol{\Sigma}}_N$, $\tilde{S}_N$ and S_N converge to the same limit in distribution, as $N \to \infty$.*

Proof of this theorem is provided in the Appendix of Raj et al. [2019]. Based on this theorem, it is tempting to ignore the additive noise and rely on the asymptotic null distribution.

However, as demonstrated in Section 5.6, such tests have a *grossly miscalibrated Type I error*, hence we propose a non-asymptotic regime in order to improve approximations of the null distribution when computing the test threshold.

In particular, let's start by recalling that we previously relied on $\sqrt{N}\mathbf{w}_N$ converging to a zero-mean multivariate normal distribution $\mathcal{N}(0, \boldsymbol{\Sigma})$, with $\boldsymbol{\Sigma} = \mathbb{E}[\mathbf{z}\mathbf{z}^\top]$ [Chwialkowski et al., 2015]. In the private setting, we will also approximate the distribution of $\sqrt{N}\tilde{\mathbf{w}}_N$ with a multivariate normal, but consider explicit non-asymptotic covariances which appear in the test statistic. Namely, the covariance of $\sqrt{N}\tilde{\mathbf{w}}_N$ is $\boldsymbol{\Sigma} + N\sigma_N^2\mathbf{I}$ and its mean is 0, so we will approximate its distribution by $\mathcal{N}(0, \boldsymbol{\Sigma} + N\sigma_N^2\mathbf{I})$. The test statistic can be understood as a squared norm of the vector $\sqrt{N}\left(\tilde{\boldsymbol{\Sigma}}_N + \gamma_N\right)^{-1/2}\tilde{\mathbf{w}}_N$. Under the normal approximation to $\sqrt{N}\tilde{\mathbf{w}}_N$ and by treating $\tilde{\boldsymbol{\Sigma}}_N$ as fixed (note that this is a quantity released to the tester), $\sqrt{N}\left(\tilde{\boldsymbol{\Sigma}}_N + \gamma_N\mathbf{I}\right)^{-1/2}\tilde{\mathbf{w}}_N$ is another multivariate normal, i.e. $\mathcal{N}(0, \mathbf{C})$, where

$$\mathbf{C} = (\tilde{\boldsymbol{\Sigma}}_N + \gamma_N\mathbf{I})^{-1/2}(\boldsymbol{\Sigma} + N\sigma_N^2\mathbf{I})(\tilde{\boldsymbol{\Sigma}}_N + \gamma_N\mathbf{I})^{-1/2}.$$

The overall statistic thus follows a distribution given by a weighted sum $\sum_{j=1}^{J}\lambda_j\chi_j^2$ of independent χ^2 distributed random variables, with the weights λ_j given by the eigenvalues of $\mathbf{C}$. Note that this approximation to the null distribution depends on a *non-private* true covariance $\boldsymbol{\Sigma}$. While that is clearly not available to the tester, we propose to simply replace this quantity with the privatised empirical covariance, i.e. $\tilde{\boldsymbol{\Sigma}}_N$, so that the tester approximates the null distribution with $\sum_{j=1}^{J}\tilde{\lambda}_j\chi_j^2$, where $\tilde{\lambda}_j$ are the eigenvalues of

$$\tilde{\mathbf{C}} = (\tilde{\boldsymbol{\Sigma}}_N + \gamma_N\mathbf{I})^{-1}(\tilde{\boldsymbol{\Sigma}}_N + N\sigma_N^2\mathbf{I}),$$

i.e. $\tilde{\lambda}_j = \frac{\tau_j + N\sigma_N^2}{\tau_j + \gamma_N}$, where $\{\tau_j\}$ are the eigenvalues of $\tilde{\boldsymbol{\Sigma}}_N$ (note that $\tilde{\lambda}_j \to 1$ as $N \to \infty$ recovering back the asymptotic null). This approach, while a heuristic, gives approximate Type I control, good power performance and is differentially private. This is unlike the approach which relies on the asymptotic null distribution and ignores the presence of privatising noise. We demonstrate this empirically in Section 5.6.

5.5.2 Trusted-curator setting: perturbed test statistic

In this section, we will consider how directly perturbing the test statistic impacts the null distribution. To achieve private test statistics, we showed that we can simply add noise using the Gaussian mechanism, described in Section 5.3.2. Similarly to Theorem 9, we have a similar theorem below, which says that the perturbed statistic then has the same asymptotic null distribution as the original statistic.

Theorem 10. *Using the noise variance $\sigma_\eta^2(\epsilon, \delta, N)$ defined by the upper bound in Theorem 8, $\tilde{S}_N$ and S_N converge to the same limit in distribution, as $N \to \infty$.*

The proof follows immediately from $\sigma_\eta(\epsilon, \delta, N) \to 0$, as $N \to \infty$. As in the case of perturbed mean and covariance, we consider approximating the null distribution with the sum of the χ^2 with J degrees of freedom and a normal $\mathcal{N}(0, \sigma_\eta^2(\epsilon, \delta, N))$, i.e., the distribution of the true statistic is approximated with its asymptotic version, whereas we use exact non-asymptotic distribution of the added noise. The test threshold can then easily be computed by a Monte Carlo test which repeatedly simulates the sum of these two random variables. It is important to note that since $\sigma_\eta^2(\epsilon, \delta, N)$ is *independent of the data* (Appendix in [Raj et al., 2019]), an untrusted tester can simulate the approximate null distribution without compromising privacy.

5.5.3 No-trusted-entity setting

Similarly as in Section 5.5.1, as $N_x, N_y \to \infty$ such that $N_x/N_y \to \rho \in (0, 1)$, asymptotic null distribution of this test statistic remains unchanged as in the non-private setting, i.e. it is the χ^2 distribution with J degrees of freedom. However, by again considering the non-asymptotic case and applying a χ^2 approximation, we get improved power and Type I control. In particular, the test statistic is close to a weighted sum $\sum_{j=1}^{J} \lambda_j \chi_j^2$ of independent χ^2 distributed random variables, with the weights λ_j given by the eigenvalues of

$$\mathbf{C} = \frac{N_x N_y}{N_x + N_y}(\tilde{\mathbf{\Sigma}}_{N_x,N_y} + \gamma_N \mathbf{I})^{-1/2}(\mathbf{\Sigma^x}/N_x + \mathbf{\Sigma^y}/N_y + (\sigma_{N_x}^2 + \sigma_{N_y}^2)\mathbf{I})(\tilde{\mathbf{\Sigma}}_{N_x,N_y} + \gamma_N \mathbf{I})^{-1/2}$$

where $\mathbf{\Sigma^x}$ and $\mathbf{\Sigma^y}$ are the true covariances within each of the samples, $\sigma_{N_x}^2$ and $\sigma_{N_y}^2$ are the variances of the noise added to the mean vectors $\mathbf{w}_{N_x}$ and $\mathbf{w}_{N_y}$, respectively. While $\mathbf{\Sigma^x}$ and

$\boldsymbol{\Sigma}^{\mathbf{y}}$ are clearly not available to the tester, the tester can replace them with their privatised empirical versions $\tilde{\boldsymbol{\Sigma}}^{\mathbf{x}}_{N_x}$ and $\tilde{\boldsymbol{\Sigma}}^{\mathbf{y}}_{N_y}$ and compute eigenvalues $\tilde{\lambda}_j$ of

$$\tilde{\mathbf{C}} = \frac{N_x N_y}{N_x + N_y}(\tilde{\boldsymbol{\Sigma}}_{N_x,N_y} + \gamma_N \mathbf{I})^{-1/2}(\tilde{\boldsymbol{\Sigma}}^{\mathbf{x}}_{N_x}/N_x + \tilde{\boldsymbol{\Sigma}}^{\mathbf{y}}_{N_y}/N_y + (\sigma^2_{N_y} + \sigma^2_{N_x})\mathbf{I})(\tilde{\boldsymbol{\Sigma}}_{N_x,N_y} + \gamma_N \mathbf{I})^{-1/2}.$$

Note that this is a differentially private quantity. Similarly as in the trusted-curator setting, we demonstrate in Section 5.6 that this corrected approximation to the null distribution leads to significant improvements in power and Type I control.

5.6 Experiments

Here we demonstrate the effectiveness of our private kernel two-sample test[1] on both synthetic and real problems, for testing $H_0 : P_X \overset{d}{=} P_Y$. The total sample size is denoted by N_{all} and the number of test set samples by N. We set the significance level to $\alpha = 0.01$. Unless specified otherwise use the isotropic Gaussian kernel with a lengthscale ℓ and fix the number of test locations to $J = 5$ (focusing on the ME test only). Under the trusted-curator (TC) setting, we use 20% of the samples N_{all} as an independent training set to optimise the test locations and ℓ using gradient descent as in Jitkrittum et al. [2016]. Under the no-trusted-entity (NTE) setting, we randomly sample J locations and calculate the median heuristic bandwidth [Gretton et al., 2012b].

For all our experiments, we average them over 500 runs, where each run repeats the simulation or randomly samples without replacement from the data set. We then report the empirical estimate of $\mathbb{P}(\tilde{S}_N > T_\alpha)$, computed by proportion of times the statistic $\tilde{S}_N$ is greater than the T_α, where T_α is the test threshold provided by the corresponding approximation to the null distribution. Regularisation parameter $\gamma = \gamma_n$ is fixed to 0.001 for TC under perturbed test statistics (TCS). In the trusted-curator mean covariance perturbation (TCMC) and NTE, given the privacy budget of (ϵ, δ), we use $(0.5\epsilon, 0.5\delta)$ to perturb the mean and covariance separately. We compare these to its non-private counterpart ME, as there are no other available appropriate baseline to compare against. We will also demonstrate the importance of using an approximated finite-null distribution versus the asymptotic null distribution.

[1] Code is available at https://github.com/hcllaw/private_tst

5.6.1 Synthetic data

We demonstrate our tests on 3 separate synthetic problems, namely, Same Gaussian (SG), Gaussian mean difference (GMD), Gaussian variance difference (GVD), with the specifications of P_X and P_Y summarised in Table 5.1. The same experimental setup was used in Jitkrittum et al. [2016].

5.6.1.1 Varying privacy level ϵ

We now fix the test sample size N to be 10000, and vary ϵ between 0 and 5 with a fixed $\delta = 1e-5$. The results are shown in the top row of Figure 5.2. For SG dataset, where $H_0 : P_X \overset{d}{=} P_Y$ is true, we can see that if one simply applies the asymptotic null distribution of a χ^2 on top, we will obtain a massively inflated Type I error. This is however not the case for TCMC, TCS and NTE, where the Type I error is approximately controlled at the right level (note that here we allow some flexibility due to multiple testing), this is shown more clearly in Figure 5.3. For the GMD and GVD dataset, the null hypothesis does not hold, and we see that our algorithms indeed discover this difference. As expected we observe a trade-off between privacy level and power, for increasing privacy (decreasing ϵ), we have less power. These experiments also reveals the order of performance of these algorithms, i.e. TCS $>$ TCMC $>$ NTE. This is not surprising, as for TCMC and NTE, we are pertubing the mean and covariance separately, rather than the statistic directly which is the direct quantity we want to protect. The power analysis for the SVD dataset also reveal the interesting nature of sampling versus optimisation in our two settings. In the SVD dataset, we observe that NTE performs better than TCS and TCMC, however if we use the same test locations and bandwidth of NTE for TCS and TCMC, the order of performance is as we expect, better for sampling over optimisation.

Data	P_X	P_Y
SG	$\mathcal{N}(\mathbf{0}, \mathbf{I}_{50})$	$\mathcal{N}(\mathbf{0}, \mathbf{I}_{50})$
GMD	$\mathcal{N}(\mathbf{0}, \mathbf{I}_{100})$	$\mathcal{N}((1, 0, \ldots, 0)^\top, \mathbf{I}_{100})$
GVD	$\mathcal{N}(\mathbf{0}, \mathbf{I}_{50})$	$\mathcal{N}(\mathbf{0}, \text{diag}(2, 1, \ldots, 1))$

Table 5.1: Synthetic problems (Null hypothesis H_0 holds only for SG). These settings are also studied in [Jitkrittum et al., 2016], Chwialkowski et al. [2015] and Gretton et al. [2012c].

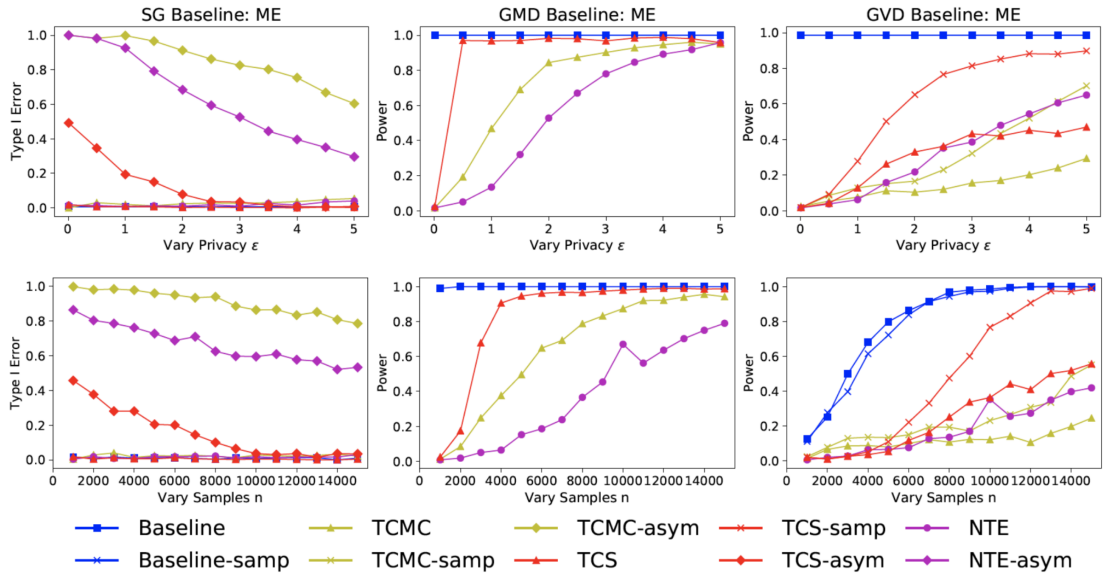

Figure 5.2: Type I error for the SG dataset, Power for the GMD, GVD dataset over 500 runs, with $\delta = 1e^{-}5$. **Top**: Varying ϵ with $N = 10000$. **Bottom**: Varying N with $\epsilon = 2.5$. Here *-asym represents using the asymptotic χ^2 null distribution, while *-samp represents sampling locations and using the median heuristic bandwidth.

5.6.1.2 Varying test sample size N

We now fix $\epsilon = 2.5$, $\delta = 1.0^{-5}$ and vary N from 1000 to 15000. The results are shown in the bottom row of Figure 5.2. The results for the SG dataset further reinforce the importance of not simply using the asymptotic null distribution, as even at very large sample size, the Type I error is still inflated when naively computing the test threshold form a χ^2 distribution. This is not the case for TCMC, TCS and NTE, where the Type I error is approximately controlled at the correct level for all sample sizes, as shown in Figure 5.3.

5.6.2 Real data: Celebrity age data

We now demonstrate our tests on a real life celebrity age dataset [Rothe et al., 2018], containing 397949 images of 19545 celebrities and their corresponding age labels. Here, we will follow the pre-processing of Law et al. [2018b], where images from the same celebrity are placed into the same bag, and the bag label is calculated as the mean age of that celebrity's images. Using these bags, we construct two datasets, under25 and 25to35, where here under25 consists of images corresponding to those bag label < 25, while the 25to35 consists

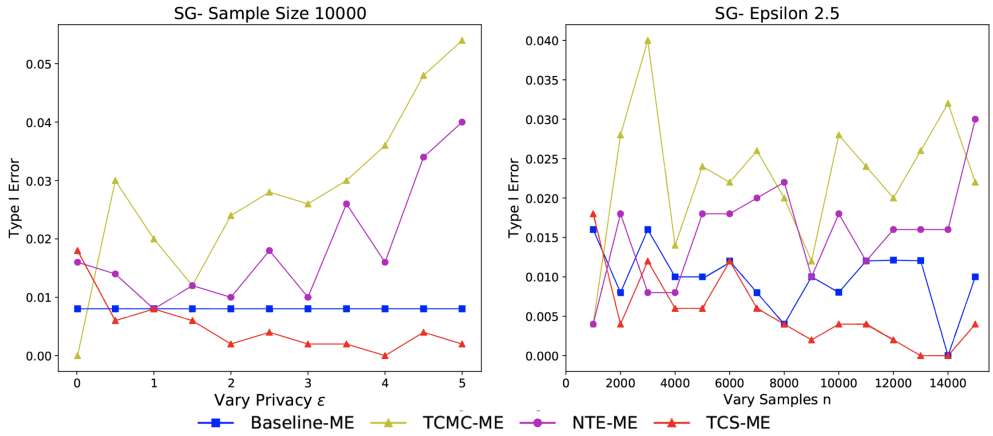

Figure 5.3: Type I error for the SG Dataset, with baselines ME, $\delta = 1e-5$. **Left:** Vary ϵ, fix $N = 10000$ **Left:** Vary N, fix $\epsilon = 2.5$.

of images with a bag label that is between 25 and 35. The dataset under25 contains 58095 images, and the dataset 25to35 contains 126415 images.

For this experiment, we will focus on using the ME version of the test and consider the kernel

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\phi(\mathbf{x}) - \phi(\mathbf{x}')||^2}{\ell}\right)$$

where $\phi(\mathbf{x}) : \mathbb{R}^{256\times256} \to \mathbb{R}^{4096}$ is the feature map learnt by the CNN in [Rothe et al., 2018], mapping the images in the original pixel space to the last layer. For our experiment, we take $N_{\text{all}} = 3125$, and use 20% of the data for sampling test locations, and calculation of the median heuristic bandwidth. Note here we do not perform optimisation, due to the large dimension of the feature map φ. We now perform two tests, for one test we compare samples from under25 only (i.e. $H_0 : P_X \overset{d}{=} P_Y$ holds), and the other we compares samples from under25 to samples from 25to35 (i.e. $H_0 : P_X \overset{d}{=} P_Y$ does not hold). The results are shown in Figure 5.4 for ϵ from 0.1 to 0.7. We observe that in the under25 only test, the TCMC, TCS and NTE all achieve the correct Type I error rate, this is unlike their counterpart that uses the χ^2 asymptotic null distribution. In the under25 vs 25to35 two-sample test, we see that our algorithms can achieve a high power (with little samples) at a high level of privacy, protecting the original images from malicious intent.

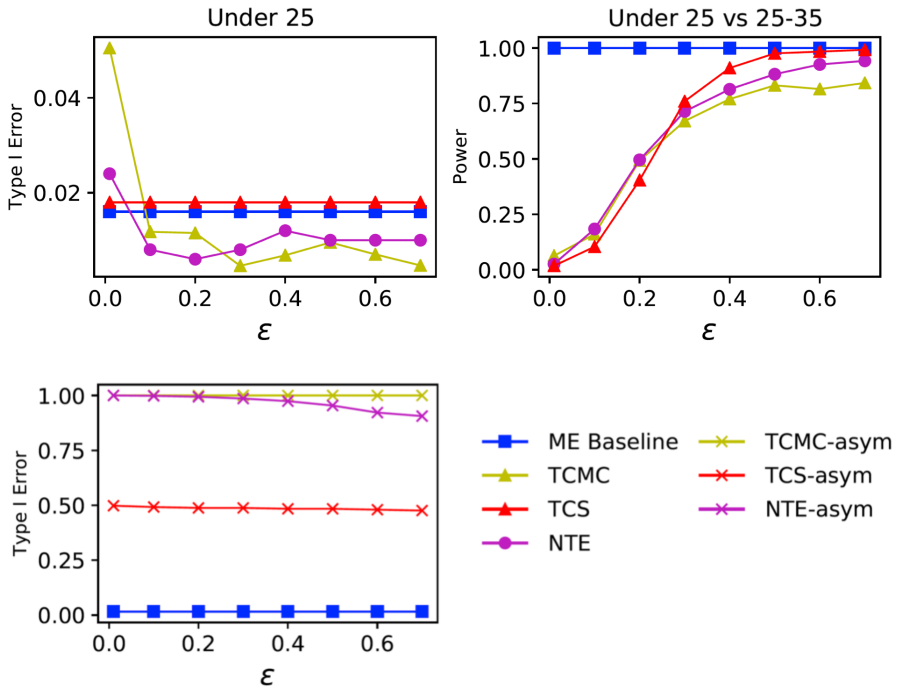

Figure 5.4: Type I error for the under25 only test, Power for the under25 vs 25to35 test over 500 runs, with $N = 2500, \delta = 1e^{-5}$. *-asym represents using the asymptotic χ^2 null distribution.

5.7 Conclusion

While kernel based hypothesis testing provides flexible statistical tools for data analysis, its utility in differentially private settings is not well understood. We investigated differentially private kernel based two-sample testing procedures, by making use of the sensitivity bounds on the quantities used in the test statistics. While asymptotic null distributions for the modified procedures remain unchanged, ignoring additive noise can lead to an inflated number of false positives. Thus, we propose new approximations of the null distributions under the private regime which give improved Type I control and good power-privacy tradeoffs, as demonstrated in extensive numerical evaluations.

5.8 Chapter appendix

5.8.1 Adding noise to data directly

To define differential privacy, we need to define two neighbouring dataset $\mathcal{D}$ and $\mathcal{D}'$. Let us consider some class of databases $\mathcal{D}^N$ where each dataset differ with another by one data

point. Suppose each database carries N data points of dimension d and that we want to release data privately with the function $f : \mathcal{D}^N \to \mathbb{R}^{nd}$ that vertically stack them in one large vector of dimension nd. Then the global sensitivity for f is:

$$GS_2(f) = \sup_{\mathcal{D},\mathcal{D}'} \|f(\mathcal{D}) - f(\mathcal{D}')\|_2 \approx \mathcal{O}(diam(\mathcal{X})) \tag{5.4}$$

where $diam(\mathcal{X})$ denotes the input space. Since the sensitive is too high (of the order of diameter of input space), the utility of the data is reduced by a huge amount after addition of noise.

Chapter 6

Discussion

In this thesis, we tackled various testing and learning problems on distributional and set inputs. In particular, by consideration and interpretation of the inherent data structure, we have constructed models that are shown to have state-of-the-art performance on various toy and real life tasks. Specifically, the following papers are presented in this thesis:

- **Ho Chung Leon Law**, Dino Sejdinovic, Ewan Cameron, Tim Lucas, Seth Flaxman, Katherine Battle, Kenji Fukumizu
  Variational learning on aggregate outputs with Gaussian processes [Law et al., 2018a]
  Advances in Neural Information Processing Systems (NeurIPS), 2018
- **Ho Chung Leon Law**, Peilin Zhao, Lucian Chan, Junzhou Huang, Dino Sejdinovic
  Hyperparameter Learning via Distributional Transfer [Law et al., 2018c]
  Advances in Neural Information Processing Systems (NeurIPS), 2019
- **Ho Chung Leon Law**, Christopher Yau, Dino Sejdinovic
  Testing and learning on distributions with symmetric noise invariance [Law et al., 2017]
  Advances in Neural Information Processing Systems (NeurIPS), 2017.
- Anant Raj*, **Ho Chung Leon Law***, Dino Sejdinovic, Mijung Park
  A Differentially Private Kernel Two-Sample Test [Raj et al., 2019]
  European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), 2019

In this thesis, we have not presented the following work:

* denote authors with equal contribution.

- **Ho Chung Leon Law***, Dougal Sutherland*, Dino Sejdinovic, and Seth Flaxman
  Bayesian Approaches to Distribution Regression [Law et al., 2018b]
  Artificial Intelligence and Statistics (AISTATS), 2018

which is related to the problem of learning from distributions, and might be of interest to the reader. Specifically, this paper suggests that current approaches do not propagate the uncertainty arising from differing bag sizes, and hence proposes a Bayesian distribution regression formalism to account for this. In particular, the paper proposes three separate Bayesian models, each targeting different uncertainties that arise in a distribution regression setting. These approaches are then compared on demonstrative toy examples, as well as a challenging problem of age prediction from a bag of face images.

6.1 Conclusion

We began our thesis by discussing the problem of learning from set inputs in Chapter 2, where the misalignment of the resolution of inputs and outputs leads to a whole set of covariates being associated with a single aggregated label. With the intention of constructing a regression function at the resolution of inputs, we have proposed a general framework of aggregated observation models using Gaussian processes, which we made tractable and scalable using variational inference. The resulting methodology not only allows for the prediction at the resolution of inputs, but it also allows for the quantification of uncertainty, which is vital in this application. In particular, we have applied our methodology to the important problem of spatial mapping of malaria, where our goal is to construct a high resolution map of the disease using only coarse incidence data and remote sensing covariates at a finer resolution. Using a real life malaria dataset with over 1 million observations and 957 aggregated labels, we empirically show that our model is able to capture known factors of malaria at the resolution of inputs, suggesting that the model can be used for the targeted delivery of malaria interventions.

In Chapter 3, we proposed a Bayesian optimisation method for the setting where a number of previous tasks have already been solved, with our goal to solve the target task. By focusing

* denote authors with equal contribution.

on tasks that are the selection of hyperparameters for machine learning models, the central idea is to extract useful meta-information from the training data to construct an appropriate similarity between tasks. Here we first construct a flexible feature map that embeds the distribution of the training data, before constructing a Gaussian process or Bayesian linear regression that jointly models all tasks. Using all the previous information available, the model can be trained in an end to end fashion, learning feature maps of distributions that can yield representations invariant to variations in the training data not relevant for hyperparameter selection. Demonstrating our approach on a range of regression and classification experiments, we show that we are able to transfer information from similar tasks, and achieve a faster convergence compared to the state-of-the-art baselines.

In Chapter 4, we focused on testing and learning scenarios with distributional inputs that are robust to symmetric additive noise. The motivation for this work is that in real life discovered differences between two distributions can be due to measurement noise, rather than the underlying distribution of interest. Hence, our goal is to design a corresponding feature map and distance that is invariant to symmetric positive definite components, including many distributions that are commonly used to model additive noise. By building on previous work in nonparametric deconvolution [Delaigle & Hall, 2016], analogous to the random Fourier features [Rahimi & Recht, 2007] and the MMD [Gretton et al., 2012a], we construct the phase features and the PhD statistic that are robust to impairment of the input distributions with common additive noise distributions. In application of the PhD and a simpler alternative approach for two-sample testing on a toy and real life Higgs Boson dataset, we discovered that indeed we can target the underlying differences of interest. Further, we showed that the phase features are also applicable to scenarios of covariate shift and robust distribution regression in an aerosol and dark matter prediction problem.

Lastly, in Chapter 5 we focus on the important problem of differential privacy in the two-sample testing, allowing for the protection of sensitive individual information. Considering two separate regimes of practical usage, we extend the current large scale kernel two-sample testing framework [Chwialkowski et al., 2015; Jitkrittum et al., 2016] to conform to differentially private constraint, through privatisation of the quantities required. As the number of data points increases, the noise required for differential privacy reduces to zero, suggesting

that the asymptotic null distribution remains unchanged. However, under a finite sample setting, empirically using the asymptotic null distribution leads to a highly miscalibrated Type I error, hence we use an approximation of the finite-sample null distribution. Through a variety of standard baseline experiments and a real life celebrity image dataset, we demonstrate that the protection of individual information is possible, while having good Type I control and good power/privacy trade-off.

6.2 Extensions

The work presented in this thesis has many potential directions for future research; here we will present a selection of these.

Ranking from team outcomes In Section 2.8 of Chapter 2, we showed that the formulation of the main model proposed in Chapter 2 can be made flexible, without explicitly having to derive any further approximations. This implies that the learning on aggregate outputs formulation with Gaussian processes can be extended to a variety of aggregation functions as inputs to any appropriate parametric family. For example, consider the problem of individual ranking of players with only observed outcomes of team games (e.g. 5v5 in say League of Legends[1]). In this case, we can treat each player as a data sample in a bag, where here the covariates are some summary statistics of the player's performance (e.g. kill-death ratio). Now by taking the output of the GP as the ability level, and summing it across all the players on the same team, we can define a Bernoulli distribution, with the probability of winning as simply the difference in the aggregated ability level of two teams (after a transformation to $[0, 1]$, e.g. logit or probit). In the case of bag overlap, the likelihood may need to be modified. Through learning with the team outcomes, and the covariates per player, the resulting learnt GP model can allow for the prediction of the ability level, given only covariates. This naturally has many applications in ranking systems in mobile and online gaming.

Multi-task hyperparameter selection As an extension to the hyperparameter learning setup in Chapter 3, we can consider the setting of multi-task hyperparameter selection [Swersky et al., 2013], whereby the goal is to simultaneously find optimal hyperparameters for all n

[1] https://euw.leagueoflegends.com/en/

tasks as quickly as possible. This differs to the current setting, as not only do we have to propose a new hyperparameter for evaluation in a given task, we also need to propose a task to evaluate this hyperparameter on. Intuitively, we would like to evaluate the hyperparameter on a selected task that will maximise the 'information' about all the tasks, hence it is vital to construct an accurate similarity matrix on tasks. Using the methodology introduced in Chapter 3, we can construct this through meta-information, before applying a multi-task acquisition function (e.g. Entropy search in Swersky et al. [2013]) to select both a task and a hyperparameter to evaluate on. The resulting methodology introduced would be useful to applications where one would like to simultaneously tune hyperparameters of many machine learning models at once (e.g. ensemble classifier).

6.3 Closing remarks

In this thesis, we have tackled various frameworks and formulations related to the machine learning models involving set and distributional inputs. Specifically, we have considered semi-supervised learning, supervised learning, meta-learning and unsupervised learning on such data structures. Importantly, throughout we have taken special notice to the importance of modelling of the data structure for the particular task in mind. For example, in Chapter 3 we designed an embedding of distributions that can learn the representations relevant for hyperparameter selection, and in Chapter 4 we constructed an embedding that removes common additive symmetric noise components. In both cases, kernel method is used as an interface to model such representations, and we believe that such methodologies will continue to gain traction, playing an important role in the modelling of flexible data structures.

Further, we have taken advantage of the growing confluence between statistical modelling and machine learning, whereby we quantify the underlying process and uncertainty in the real world with statistical models, while using performant machine learning techniques and architectures to capture the complex underlying structure. As an example in Chapter 2, we designed a bespoke statistical model to represent the process of malaria incidences, while using a Gaussian process to model the complex dependencies on the covariates. Such utilisation and understanding into their connections will be important in creation of future research directions.

Lastly, we believe that machine learning will remain crucial to human development in many years to come. In particular, the understanding into how to utilise the data in an appropriate fashion will play an important role in making better machine learning models and pushing boundaries in multiple fields. As a first step in this thesis, we have made progress towards understanding some of these models, datasets and applications, and we believe that this will provide a platform that future researchers can build on.

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, (pp. 265–283).

Ancarani, L., & Gasaneo, G. (2008). Derivatives of any order of the confluent hypergeometric function f 1 1 (a, b, z) with respect to the parameter a or b. *Journal of Mathematical Physics*, *49*(6), 063508.

Baldi, P., Sadowski, P., & Whiteson, D. (2014). Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, *5*.

Balle, B., & Wang, Y.-X. (2018). Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*, (pp. 403–412).

Balog, M., Tolstikhin, I., & Schölkopf, B. (2017). Differentially Private Database Release via Kernel Mean Embeddings. ArXiv: 1710.01641.
URL `http://arxiv.org/abs/1710.01641`

Bardenet, R., Brendel, M., Kégl, B., & Sebag, M. (2013). Collaborative hyperparameter tuning. In *International Conference on Machine Learning*, (pp. 199–207).

Belkin, M., Niyogi, P., & Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, *7*(Nov), 2399–2434.

Berlinet, A., & Thomas-Agnan, C. (2011). *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media.

Bhatt, S., Weiss, D., Cameron, E., Bisanzio, D., Mappin, B., Dalrymple, U., Battle, K., Moyes, C., Henry, A., Eckhoff, P., et al. (2015). The effect of malaria control on plasmodium falciparum in africa between 2000 and 2015. *Nature*, *526*(7572), 207.

Bishop, C. (2006). *Pattern recognition and machine learning*. Springer New York.

Blanchard, G., Deshmukh, A. A., Dogan, U., Lee, G., & Scott, C. (2017). Domain generalization by marginal transfer learning. *arXiv preprint arXiv:1711.07910*.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, *112*(518), 859–877.

Borgwardt, K. M., Gretton, A., Rasch, M. J., Kriegel, H.-P., Schölkopf, B., & Smola, A. J. (2006). Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, *22*(14), e49–e57.
URL http://dx.doi.org/10.1093/bioinformatics/btl242

Bouchard, M., Jousselme, A.-L., & Doré, P.-E. (2013). A proof for the positive definiteness of the jaccard index matrix. *International Journal of Approximate Reasoning*, *54*(5), 615–626.

Brochu, E., Cora, V. M., & De Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.

Chan, L., Hutchison, G. R., & Morris, G. M. (2019). Bayesian optimization for conformer generation. *Journal of Cheminformatics*, *11*(1), 32.
URL https://doi.org/10.1186/s13321-019-0354-7

Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., Xiao, T., Xu, B., Zhang, C., & Zhang, Z. (2015). Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274*.

Cheplygina, V., Tax, D. M., & Loog, M. (2015). On classification with bags, groups and sets. *Pattern Recognition Letters*, *59*, 11 – 17.
URL http://www.sciencedirect.com/science/article/pii/S0167865515000793

Christmann, A., & Steinwart, I. (2010). Universal kernels on non-standard input spaces. In *Advances in neural information processing systems*, (pp. 406–414).

Chwialkowski, K. P., Ramdas, A., Sejdinovic, D., & Gretton, A. (2015). Fast two-sample testing with analytic representations of probability measures. In *Advances in Neural Information Processing Systems*, (pp. 1981–1989).

Dai, H., Umarov, R., Kuwahara, H., Li, Y., Song, L., & Gao, X. (2017). Sequence2vec: a novel embedding approach for modeling transcription factor binding affinity landscape. *Bioinformatics*, *33*(22), 3575–3583.

Delaigle, A., & Hall, P. (2016). Methodology for non-parametric deconvolution when the error distribution is unknown. *Journal of the Royal Statistical Society: Series B*.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., & Naor, M. (2006a). Our data, ourselves: Privacy via distributed noise generation. In *Eurocrypt*, vol. 4004, (pp. 486–503). Springer.

Dwork, C., McSherry, F., Nissim, K., & Smith, A. (2006b). Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, (pp. 265–284). Springer.

Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, *9*, 211–407.
URL http://dx.doi.org/10.1561/0400000042

Dwork, C., Talwar, K., Thakurta, A., & Zhang, L. (2014). Analyze Gauss: optimal bounds for privacy-preserving principal component analysis. In *Symposium on Theory of Computing, STOC 2014*, (pp. 11–20).
URL http://doi.acm.org/10.1145/2591796.2591883

Feurer, M., Letham, B., & Bakshy, E. (2018). Scalable meta-learning for bayesian optimization using ranking-weighted gaussian process ensembles. In *AutoML Workshop at ICML*.

Feurer, M., Springenberg, J. T., & Hutter, F. (2014). Using meta-learning to initialize bayesian optimization of hyperparameters. In *Proceedings of the 2014 International Conference on Meta-learning and Algorithm Selection-Volume 1201*, (pp. 3–10). Citeseer.

Feurer, M., Springenberg, J. T., & Hutter, F. (2015). Initializing bayesian hyperparameter optimization via meta-learning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Flaxman, S., Sutherland, D. J., Wang, Y.-X., & Teh, Y.-W. (2016). Understanding the 2016 US presidential election using ecological inference and distribution regression with census microdata. arXiv:1611.03787.

Flaxman, S., Wang, Y.-X., & Smola, A. J. (2015). Who Supported Obama in 2012?: Ecological inference through distribution regression. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (pp. 289–298). ACM.

Fukumizu, K., Song, L., & Gretton, A. (2013). Kernel bayes' rule: Bayesian inference with positive definite kernels. *The Journal of Machine Learning Research*, *14*(1), 3753–3783.

Gaboardi, M., Lim, H. W., Rogers, R., & Vadhan, S. P. (2016). Differentially private chi-squared hypothesis testing: Goodness of fit and independence testing. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, (pp. 2111–2120).

Gaboardi, M., & Rogers, R. M. (2017). Local private hypothesis testing: Chi-square tests. *CoRR*, *abs/1709.07155*.
URL `http://arxiv.org/abs/1709.07155`

Gaulton, A., Hersey, A., Nowotka, M., Bento, A. P., Chambers, J., Mendez, D., Mutowo, P., Atkinson, F., Bellis, L. J., Cibrián-Uhalte, E., et al. (2016). The chembl database in 2017. *Nucleic acids research*, *45*(D1), D945–D954.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian data analysis*. CRC press.

Geman, S., & Geman, D. (1987). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. In *Readings in computer vision*, (pp. 564–584). Elsevier.

Gething, P. W., Casey, D. C., Weiss, D. J., Bisanzio, D., Bhatt, S., Cameron, E., Battle, K. E., Dalrymple, U., Rozier, J., Rao, P. C., et al. (2016). Mapping plasmodium falciparum

mortality in africa between 1990 and 2015. *New England Journal of Medicine*, *375*(25), 2435–2445.

Gomes, T. A., Prudêncio, R. B., Soares, C., Rossi, A. L., & Carvalho, A. (2012). Combining meta-learning and search techniques to select parameters for support vector machines. *Neurocomputing*, *75*(1), 3–13.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Goovaerts, P. (2010). Combining areal and point data in geostatistical interpolation: Applications to soil science and medical geography. *Mathematical Geosciences*, *42*(5), 535–554. URL `https://doi.org/10.1007/s11004-010-9286-5`

Goyal, V., Khurana, D., Mironov, I., Pandey, O., & Sahai, A. (2016). Do distributed differentially-private protocols require oblivious transfer?. In *ICALP*, (pp. 29:1–29:15).

Gretton, A. (2015). Notes on mean embeddings and covariance operators.

Gretton, A., Borgwardt, K. M., Rasch, M., Schölkopf, B., & Smola, A. J. (2007). A kernel method for the two-sample-problem. In B. Schölkopf, J. C. Platt, & T. Hoffman (Eds.) *NeurIPS*, (pp. 513–520). MIT Press.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. (2012a). A kernel two-sample test. *Journal of Machine Learning Research*, *13*(Mar), 723–773.

Gretton, A., Fukumizu, K., Harchaoui, Z., & Sriperumbudur, B. K. (2009). A fast, consistent kernel two-sample test. In *NeurIPS*, (pp. 673–681).

Gretton, A., Sriperumbudur, B., Sejdinovic, D., Balakrishnan, S., Pontil, M., Fukumizu, K., et al. (2012b). Optimal kernel choice for large-scale two-sample tests. In *NeurIPS 25*, (pp. 1214–1222).

Gretton, A., Sriperumbudur, B. K., Sejdinovic, D., Strathmann, H., Balakrishnan, S., Pontil, M., & Fukumizu, K. (2012c). Optimal kernel choice for large-scale two-sample tests. In *NeurIPS*.

Hall, R., Rinaldo, A., & Wasserman, L. (2013). Differential privacy for functions and functional data. *JMLR*, *14*(Feb), 703–727.

Hamelijnck, O., Damoulas, T., Wang, K., & Girolami, M. (2019). Multi-resolution multi-task gaussian processes. *arXiv preprint arXiv:1906.08344*.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications.

Haußmann, M., Hamprecht, F. A., & Kandemir, M. (2017). Variational bayesian multiple instance learning with gaussian processes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 6570–6579).

Hensman, J., Fusi, N., & Lawrence, N. D. (2013). Gaussian processes for big data. In *Uncertainty in Artificial Intelligence*, (p. 282). Citeseer.

Hensman, J., Matthews, A., & Ghahramani, Z. (2015). Scalable Variational Gaussian Process Classification. In G. Lebanon, & S. V. N. Vishwanathan (Eds.) *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, vol. 38 of *Proceedings of Machine Learning Research*, (pp. 351–360). San Diego, California, USA: PMLR.
URL http://proceedings.mlr.press/v38/hensman15.html

Hernández-Lobato, J. M., Hoffman, M. W., & Ghahramani, Z. (2014). Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems*, (pp. 918–926). Cambridge, MA, USA: MIT Press.

Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, *14*(1), 1303–1347.
URL http://dl.acm.org/citation.cfm?id=2567709.2502622

Homer, N., Szelinger, S., Redman, M., Duggan, D., Tembe, W., Muehling, J., Pearson, J. V., Stephan, D. A., Nelson, S. F., & Craig, D. W. (2008). Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLOS Genetics*, *4*(8), 1–9.
URL https://doi.org/10.1371/journal.pgen.1000167

Howitt, R., & Reynaud, A. (2003). Spatial disaggregation of agricultural production data using maximum entropy. *European Review of Agricultural Economics*, *30*(3), 359–387.
URL http://dx.doi.org/10.1093/erae/30.3.359

Hutter, F., Kotthoff, L., & Vanschoren, J. (Eds.) (2019). *Automatic Machine Learning: Methods, Systems, Challenges*. Springer.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, (pp. 448–456).

Jain, P., & Thakurta, A. (2013). Differentially private learning with kernels. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, (pp. 118–126).
URL https://www.microsoft.com/en-us/research/publication/differentially-private-learning-with-kernels/

Jitkrittum, W., Szabó, Z., Chwialkowski, K., & Gretton, A. (2016). Interpretable distribution features with maximum testing power. In *NeurIPS*.

Johnson, A., & Shmatikov, V. (2013). Privacy-preserving data exploration in genome-wide association studies. In *ACM SIGKDD 2013*.

Jolliffe, I. (2011). *Principal component analysis*. Springer.

Jones, E., Oliphant, T., Peterson, P., et al. (2001–). SciPy: Open source scientific tools for Python.
URL http://www.scipy.org/

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, *37*(2), 183–233.

Keil, P., Belmaker, J., Wilson, A. M., Unitt, P., & Jetz, W. (2013). Downscaling of species distribution models: a hierarchical approach. *Methods in Ecology and Evolution*, *4*(1), 82–94.

Kim, J., Kim, S., & Choi, S. (2018). Learning to warm-start bayesian hyperparameter optimization. *arXiv preprint arXiv:1710.06219*.

Kingma, D., & Ba, J. (2015). Adam: A method for stochastic optimization. In *ICLR*. arXiv:1412.6980.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Klein, A., Falkner, S., Bartels, S., Hennig, P., & Hutter, F. (2017). Fast bayesian optimization of machine learning hyperparameters on large datasets. In *Artificial Intelligence and Statistics*, (pp. 528–536).

Klypin, A., Yepes, G., Gottlober, S., Prada, F., & Hess, S. (2014). MultiDark simulations: the story of dark matter halo concentrations and density profiles. arXiv:1411.4001.

Kotzias, D., Denil, M., De Freitas, N., & Smyth, P. (2015). From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (pp. 597–606). ACM.

Kueck, H., & de Freitas, N. (2005). Learning about individuals from group statistics. In *21st Uncertainty in Artificial Intelligence (UAI)*, (pp. 332–339).

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, *22*(1), 79–86.

Law, H. C. L., Sejdinovic, D., Cameron, E., Lucas, T. C., Flaxman, S., Battle, K., & Fukumizu, K. (2018a). Variational learning on aggregate outputs with gaussian processes. *NeurIPS*.

Law, H. C. L., Sutherland, D., Sejdinovic, D., & Flaxman, S. (2018b). Bayesian approaches to distribution regression. In *International Conference on Artificial Intelligence and Statistics*, (pp. 1167–1176).

Law, H. C. L., Yau, C., & Sejdinovic, D. (2017). Testing and learning on distributions with symmetric noise invariance. In *NeurIPS*.

Law, H. C. L., Zhao, P., Chan, L., Huang, J., & Sejdinovic, D. (2018c). Hyperparameter learning via distributional transfer. *arXiv preprint arXiv:1810.06305*.

Lichman, M. (2013). UCI machine learning repository.
URL http://archive.ics.uci.edu/ml

Linnik, Y. V., & Ostrovskii, I. (1977). *Decomposition of random variables and vectors*.

Lloyd, C., Gunter, T., Osborne, M., & Roberts, S. (2015). Variational inference for gaussian process modulated poisson processes. In *International Conference on Machine Learning*, (pp. 1814–1822).

McGregor, A., Mironov, I., Pitassi, T., Reingold, O., Talwar, K., & Vadhan, S. (2010). The limits of two-party differential privacy. In *IEEE*.

McLeod, M., Osborne, M. A., & Roberts, S. J. (2018). Optimization, fast and slow: optimally switching between local and Bayesian optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*.
URL http://arxiv.org/abs/1805.08610

Melnikov, V., & Hüllermeier, E. (2016). Learning to aggregate using uninorms. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, (pp. 756–771). Springer.

Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (1994). *Machine learning, neural and statistical classification*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, (pp. 3111–3119).

Močkus, J. (1975). On bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference*, (pp. 400–404). Springer.

Muandet, K., Fukumizu, K., Dinuzzo, F., & Schölkopf, B. (2012). Learning from distributions via support measure machines. In *Advances in Neural Information Processing Systems 25*, (pp. 10–18).

Muandet, K., Fukumizu, K., Sriperumbudur, B., & Schölkopf, B. (2017). Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine*

Learning, *10*(1-2), 1–141.
URL http://dx.doi.org/10.1561/2200000060

Musicant, D. R., Christensen, J. M., & Olson, J. F. (2007). Supervised learning by training on aggregate outputs. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, (pp. 252–261). IEEE.

Narayanan, A., Chandramohan, M., Venkatesan, R., Chen, L., Liu, Y., & Jaiswal, S. (2017). graph2vec: Learning distributed representations of graphs. *arXiv preprint arXiv:1707.05005*.

Neal, R. M., et al. (2011). Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, *2*(11), 2.

Nickisch, H., & Rasmussen, C. (2008). Approximations for binary gaussian process classification. *Journal of Machine Learning Research*, *9*, 2035–2078.

Ntampaka, M., Trac, H., Sutherland, D. J., Battaglia, N., Póczos, B., & Schneider, J. (2015). A machine learning approach for dynamical mass measurements of galaxy clusters. *The Astrophysical Journal*, *803*(2), 50. arXiv:1410.0686.

Ntampaka, M., Trac, H., Sutherland, D. J., Fromenteau, S., Poczos, B., & Schneider, J. (2016). Dynamical mass measurements of contaminated galaxy clusters using machine learning. *The Astrophysical Journal*, *831*(2), 135. arXiv:1509.05409.

Oh, C., Gavves, E., & Welling, M. (2018). Bock: Bayesian optimization with cylindrical kernels. *arXiv preprint arXiv:1806.01619*.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. (2017). Automatic differentiation in pytorch. In *NeurIPS-W*.

Patrini, G., Nock, R., Caetano, T., & Rivera, P. (2014). (almost) no label no cry. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Weinberger (Eds.) *Advances in Neural Information Processing Systems 27*, (pp. 190–198). Curran Associates, Inc.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D.,

Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Perrone, V., Jenatton, R., Seeger, M. W., & Archambeau, C. (2018). Scalable hyperparameter transfer learning. In *Advances in Neural Information Processing Systems*, (pp. 6846–6856).

Pfahringer, B., Bensusan, H., & Giraud-Carrier, C. G. (2000). Meta-learning by landmarking various learning algorithms. In *ICML*.

Poloczek, M., Wang, J., & Frazier, P. I. (2016). Warm starting bayesian optimization. In *Proceedings of the 2016 Winter Simulation Conference*, (pp. 770–781). IEEE Press.

Quadrianto, N., Smola, A. J., Caetano, T. S., & Le, Q. V. (2009). Estimating labels from label proportions. *JMLR*, *10*, 2349–2374.

Quiñonero Candela, J., & Rasmussen, C. E. (2005). A unifying view of sparse approximate gaussian process regression. *J. Mach. Learn. Res.*, *6*, 1939–1959.

Quinonero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. D. (2009). *Dataset Shift in Machine Learning*. The MIT Press.

Rahimi, A., & Recht, B. (2007). Random features for large-scale kernel machines. In *Advances in neural information processing systems*, (pp. 1177–1184).

Raj, A., Law, H. C. L., Sejdinovic, D., & Park, M. (2019). A differentially private kernel two-sample test. *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*.

Ralaivola, L., Swamidass, S. J., Saigo, H., & Baldi, P. (2005). Graph kernels for chemical informatics. *Neural networks*, *18*(8), 1093–1110.

Rasmussen, C. E., & Williams, C. K. (2006). Gaussian processes for machine learning.

Reif, M., Shafait, F., & Dengel, A. (2012). Meta-learning for evolutionary parameter optimization of classifiers. *Machine learning*, *87*(3), 357–380.

Rezende, D. J., & Mohamed, S. (2015). Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*.

Rogers, R., & Kifer, D. (2017). A new class of private chi-square hypothesis tests. In *Artificial Intelligence and Statistics*, (pp. 991–1000).

Rossberg, H.-J. (1995). Positive definite probability densities and probability distributions. *Journal of Mathematical Sciences*, *76*(1), 2181–2197.

Rothe, R., Timofte, R., & Van Gool, L. (2018). Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, (pp. 1–14).

Rudin, W. (1962). *Fourier analysis on groups*, vol. 121967. Wiley Online Library.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels: support vector machines, regularization, optimization and beyond*. the MIT Press.

Smola, A., Gretton, A., Song, L., & Schölkopf, B. (2007). A hilbert space embedding for distributions. In *Algorithmic Learning Theory*, (pp. 13–31). Springer.

Smola, A. J., & Bartlett, P. L. (2001). Sparse greedy gaussian process regression. In *Advances in neural information processing systems*, (pp. 619–625).

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, (pp. 2951–2959).

Song, L., Fukumizu, K., & Gretton, A. (2013). Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *Signal Processing Magazine, IEEE*, *30*(4), 98–111.

Springenberg, J. T., Klein, A., Falkner, S., & Hutter, F. (2016). Bayesian optimization with robust bayesian neural networks. In *Advances in Neural Information Processing Systems*, (pp. 4134–4142).

Srinivas, N., Krause, A., Kakade, S. M., & Seeger, M. (2009). Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*.

Sriperumbudur, B. K., Gretton, A., Fukumizu, K., Schölkopf, B., & Lanckriet, G. R. (2010). Hilbert space embeddings and metrics on probability measures. *JMLR*, *99*, 1517–1561.

Steinwart, I., & Christmann, A. (2008). *Support Vector Machines*. Springer Publishing Company, Incorporated, 1st ed.

Sutherland, D. J. (2016). *Scalable, Flexible, and Active Learning on Distributions*. Ph.D. thesis, Carnegie Mellon University.

Sutherland, D. J., Oliva, J. B., Póczos, B., & Schneider, J. G. (2016). Linear-time learning on distributions with approximate kernel embeddings. In *Proc. AAAI Conference on Artificial Intelligence*, (pp. 2073–2079).

Swersky, K., Snoek, J., & Adams, R. P. (2013). Multi-task bayesian optimization. In *Advances in neural information processing systems*, (pp. 2004–2012).

Szabó, Z., Sriperumbudur, B. K., Póczos, B., & Gretton, A. (2016). Learning theory for distribution regression. *The Journal of Machine Learning Research*, *17*(1), 5272–5311.

Teh, Y. W., Newman, D., & Welling, M. (2007). A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In *Advances in neural information processing systems*, (pp. 1353–1360).

Titsias, M. (2009). Variational learning of inducing variables in sparse gaussian processes. In D. van Dyk, & M. Welling (Eds.) *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, vol. 5 of *Proceedings of Machine Learning Research*, (pp. 567–574). Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR. URL `http://proceedings.mlr.press/v5/titsias09a.html`

Todorovski, L., Brazdil, P., & Soares, C. (2000). Report on the experiments with feature selection in meta-level learning. In *Proceedings of the PKDD-00 workshop on data mining, decision support, meta-learning and ILP: forum for practical problem presentation and prospective solutions*. Citeseer.

Vanschoren, J., van Rijn, J. N., Bischl, B., & Torgo, L. (2013). Openml: Networked science in machine learning. *SIGKDD Explorations*, *15*(2), 49–60.
URL `http://doi.acm.org/10.1145/2641190.2641198`

Wahba, G. (1990). *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics.

Wang, J., Clark, S. C., Liu, E., & Frazier, P. I. (2016). Parallel bayesian global optimization of expensive functions. *arXiv preprint arXiv:1602.05149*.

Wang, Z., Lan, L., & Vucetic, S. (2012). Mixture model for multiple instance regression and applications in remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, *50*(6), 2226–2237.

Warrel, D., Cox, T., Firth, J., & Benz, J. E. (2017). Oxford textbook of medicine.

Weiss, D. J., Mappin, B., Dalrymple, U., Bhatt, S., Cameron, E., Hay, S. I., & Gething, P. W. (2015). Re-examining environmental correlates of plasmodium falciparum malaria endemicity: a data-intensive variable selection approach. *Malaria journal*, *14*(1), 68.

Wendland, H. (2004). *Scattered Data Approximation*. Cambridge, UK: Cambridge University Press.

Wilson, A. G., Hu, Z., Salakhutdinov, R., & Xing, E. P. (2016). Deep kernel learning. In *Artificial Intelligence and Statistics*, (pp. 370–378).

Wistuba, M., Schilling, N., & Schmidt-Thieme, L. (2018). Scalable gaussian process-based transfer surrogates for hyperparameter optimization. *Machine Learning*, *107*(1), 43–78.

Xavier, A., de Belém Costa Freitas, M., do Socorro Rosário, M., & Fragoso, R. (2018). Disaggregating statistical data at the field level: An entropy approach. *Spatial Statistics*, *23*, 91 – 108.
URL http://www.sciencedirect.com/science/article/pii/S2211675317301707

Yu, F. X., Choromanski, K., Kumar, S., Jebara, T., & Chang, S.-F. (2014). On learning from label proportions. *arXiv preprint arXiv:1402.5902*.

Yu, F. X., Liu, D., Kumar, S., Jebara, T., & Chang, S.-F. (2013). $\propto$ svm for learning with label proportions. *arXiv preprint arXiv:1306.0886*.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R., & Smola, A. (2017). Deep sets. In *NeurIPS*.