Sparse Distance Metric Learning

Tze Leung Choy

St John's College

University of Oxford

Doctor of Philosophy

Michaelmas 2014

Abstract

A good distance metric can improve the accuracy of a nearest neighbour classifier. Xing et al. (2002) proposed distance metric learning to find a linear transformation of the data so that observations of different classes are better separated. For high-dimensional problems where many uninformative variables are present, it is attractive to select a sparse distance metric, both to increase predictive accuracy but also to aid interpretation of the result. In this thesis, we investigate three different types of sparsity assumption for distance metric learning and show that sparse recovery is possible under each type of sparsity assumption with an appropriate choice of $\ell 1$-type penalty. We show that a lasso penalty promotes learning a transformation matrix having lots of zero entries, a group lasso penalty recovers a transformation matrix having zero rows/columns and a trace norm penalty allows us to learn a low rank transformation matrix. The regularization allows us to consider a large number of covariates and we apply the technique to an expanded set of basis called rule ensemble to allow for a more flexible fit. Finally, we illustrate an application of the metric learning problem via a document retrieval example and discuss how similarity-based information can be applied to learn a classifier.

Acknowledgements

First of all, I would like to express thanks to the University of Oxford Clarendon Fund Scholarship for their financial support. The completion of this dissertation is only possible with support from many people. I would like to express my greatest gratitude to Professor Nicolai Meinshausen for his guidance as my supervisor. I would also like to thank the support staff in the Department of Statistics, especially the computing officers for solving numerous technical problems. Finally, I would like to thank my parents, my sister and my loved one for their support and encouragement.

Contents

Chapter 1

Introduction

Data classification algorithms such as k-nearest neighbour can often benefit from learning a distance metric. Distance metric learning is a term coined by Xing et al. (2002) to find a Mahalanobis distance such that distances between observations of the same class are small and distances between observations of different classes are large. A number of metric learning algorithms have been proposed. In this chapter, we first introduce Mahalanobis distance metric learning and review a number of existing metric learning algorithms. We discuss why sparsity can potentially improve the estimation and introduce sparse metric learning using an $\ell 1$ regularization.

The organization of this chapter is as followed. We first introduce the problem of Mahalanobis distance metric learning and its connection with learning a linear transformation of the data in Section 1.1 and review existing algorithms that aim at improving nearest neighbour accuracy in Section 1.2. In Section 1.3, we propose sparse metric learning. We highlight some of our contributions in Section 1.4. Finally, we outline the contents of the following chapters in Section 1.5.

1.1 Mahalanobis distance metric learning

Let $\mathbf{X}$ be a $n \times p$ design matrix where each row corresponds to one observation while each column corresponds to each of the p variables. Let x_i^T be the i$^{\text{th}}$ row of the matrix X which corresponds to the i$^{\text{th}}$ observation.

The Mahalanobis distance under a positive semi-definite matrix $\mathbf{M}$ is defined as

$$d_{\mathbf{M}}(i,j) = (x_i - x_j)^T \mathbf{M}(x_i - x_j) \text{ where } \mathbf{M} \text{ is a positive semi-definite matrix.} \quad (1.1)$$

The problem of distance metric learning is to find a positive semi-definite matrix $\mathbf{M}$ such that the resulting distances improve distance-based classifier accuracy.

An alternative explanation of the metric learning problem is that we are trying to find a transformation of the data. Consider the Cholesky decomposition of $\mathbf{M}$, $\mathbf{M} = \mathbf{A}^T\mathbf{A}$ where $\mathbf{A}$ is an upper triangular matrix.

We can then rewrite the distances formulae in terms of $\mathbf{A}$.

$$\begin{aligned} d_{\mathbf{M}}(i,j) &= (x_i - x_j)^T\mathbf{M}(x_i - x_j) && (1.2)\\ &= (x_i - x_j)^T\mathbf{A}^T\mathbf{A}(x_i - x_j) && (1.3)\\ &= (\mathbf{A}x_i - \mathbf{A}x_j)^T(\mathbf{A}x_i - \mathbf{A}x_j) && (1.4) \end{aligned}$$

Therefore, finding a positive semi-definite matrix $\mathbf{M} = \mathbf{A}^T\mathbf{A}$ is equivalent to finding a transformation matrix $\mathbf{A}$. Working with $\mathbf{M}$ avoids problem with non-identifiability associated with finding the transformation matrix $\mathbf{A}$, for example, the matrices $\mathbf{A}$ and $-\mathbf{A}$ define the same metric.

1.2 Metric learning for classification

We discuss several metric learning methods that aim at improving nearest neighbour classification in this section. Since directly optimizing the predictive accuracy is non-smooth and non-convex optimization, the metric is trained indirectly by minimizing an alternative smooth loss function. Different loss functions have been proposed. Xing et al. (2002) and Weinberger and Saul (2009) try to keep observations of the same class close together while separating observations of different classes. Goldberger et al. (2008) tries to directly optimize the classification accuracy.

1.2.1 Formulation by Xing et al.

Our investigation in distance metric learning is motivated by Xing et al. (2002). The key idea is to keep all the points of the same class close together while separating observations of different classes as far as possible from each other.

They formulate the metric learning problem as the following optimization. Let

$$y_{i,j} = \begin{cases} 1 & \text{if observations i and j are of different classes} \\ 0 & \text{otherwise} \end{cases} \quad (1.5)$$

and propose to find $\hat{\mathbf{M}}$ by

$$\text{argmax}_{\mathbf{M}} \sum_{i,j} y_{i,j} \sqrt{d_{\mathbf{M}}(i,j)}, \tag{1.6}$$

$$\text{such that } \mathbf{M} \succeq 0 \text{ and } \sum_{i,j} (1 - y_{i,j}) d_{\mathbf{M}}(i,j) \leq 1. \tag{1.7}$$

Maximizing (1.6) separates observations of different classes while constraint (1.7) keeps points of the same class close together. This problem is convex and optimization was performed by gradient ascent and iterative projection back to the cone of positive definite matrices.

Inspired by Xing et al. (2002), a lot of new distance metric learning algorithms were proposed to improve classification accuracy of nearest neighbour. We now discuss two of them, Neighbourhood Component Analysis (Goldberger et al., 2008) and Large Margin Nearest Neighbour (Weinberger and Saul, 2009).

1.2.2 Neighbourhood Component Analysis

Neighbourhood component analysis was proposed by Goldberger et al. (2008). This method is motivated by stochastic neighbour assignment. For each observation i, we select a neighbour j of i randomly according to the probability p_{ij} and assign the label of j as the prediction of i.

The probability p_{ij} is defined as

$$\begin{aligned} p_{ij} &= \frac{\exp(-d_{\mathbf{A}}(i,j))}{\sum_k \exp(-d_{\mathbf{A}}(i,k))} \text{ for } i \neq j, \\ p_{ii} &= 0. \end{aligned}$$

where we would like to learn a $p \times p$ transformation matrix $\mathbf{A}$ with pairwise distance defined as

$$d_{\mathbf{A}}(i,j) = (\mathbf{A}x_i - \mathbf{A}x_j)^T(\mathbf{A}x_i - \mathbf{A}x_j).$$

The probability p_i that point i is correctly classified would be

$$p_i = \sum_{j:y_{i,j}=0} p_{i,j}.$$

The objective is to maximize the expected number of points correctly classified

$$\hat{\mathbf{A}} = \text{argmax}_{\mathbf{A}} \sum_i p_i.$$

The objective tries to maximize the probability that a point is correctly classified directly. However, the above optimization problem is not convex and has local maxima. We might need to start the algorithm with different initial values to find better local maxima.

1.2.3 Large Margin Nearest Neighbour

Weinberger and Saul (2009) proposed the large margin nearest neighbour. Let N be the set of target neighbours, $N = \{(i,j) : y_{i,j} = 0\}$ and let D be the set of all pairs with different class labels, $D = \{(i,j) : y_{i,j} = 1\}$. The initial target neighbours is often chosen to be k-nearest neighbours of the same class using Euclidean distances instead of all observations of the same classes.

The motivation of their method is to find a metric $\mathbf{M}$ such that all target neighbours are closer to each other than observations of different classes with a margin of size one. That is, we would like to find a distance that satisfy the following relation.

$$\forall\{(i,j,k) : (i,j) \in N, (i,k) \in D\}, \quad d(i,j) + 1 \leq d(i,k)$$

The loss function they propose combine a term that pulls all target neighbours as close as possible and maintain the above margin condition using a hinge loss formulation.

$$L(\mathbf{M}) = \sum_{(i,j)\in N} d(i,j) + \sum_{(i,j)\in N,(i,k)\in D} [d(i,j) - d(i,k) + 1]_+$$

where

$$[x]_+ = \begin{matrix} x & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{matrix} .$$

The main advantage of this formulation is that it is a convex optimization problem and it has good performance in various classification tasks.

1.2.4 Other related topics of metric learning

1.2.4.1 Linear discriminant analysis

Linear discriminant analysis attempts to find a projection of the data that maximize the ratio of between class variance and within class variance. The set of linear discriminants naturally define a projection onto a space of dimension $K-1$ where K is the number of classes. The first few linear discriminants often give useful low dimensional representation of the data. We refer to Ripley (1996), Venables and Ripley (2002) and Hastie et al. (2003) for details. Hastie and Tibshirani (1996) suggested a local adaptive learning algorithm based on a modification of Linear Discriminant Analysis by computing the between and within classes variances only with observations near the test point.

1.2.4.2 Unsupervised methods

We have mainly focused on supervised metric learning: methods that uses the class labels of the observations. Unsupervised distance metric learning is another closely related topic which we shall describe briefly here.

Principal component analysis finds a lower dimensional linear projection of the data that gives maximal variability. It is a widely used classical method in dimension reduction. It can be computed easily by picking the eigenvectors associated with the largest eigenvalues of the covariance matrix. Multidimensional scaling finds an embedding of the data that preserves the pairwise distance. Multidimensional scaling is equivalent to principal component analysis when the pairwise distance is given by the Euclidean metric.

Tenenbaum et al. (2000) suggested using geodesic manifold distances instead of Euclidean distances when applying multidimensional scaling. Geodesic distance allows us to discover non-linear manifolds of the data which classical methods based on Euclidean distance fail to recognize. To compute the geodesic distance, we first construct a neighbourhood graph by joining up the k-nearest neighbour. We can then approximate the geodesic distance between two points by finding the shortest path on the neighbourhood graph.

If we assume that the neighbourhood of each point is locally linear, then each observation can be approximated by the weighted average of points nearby. Roweis and Saul (2000) considered a lower dimensional embedding that preserves these local structures. We first find a weight matrix W^* that minimizes $\sum_i ||x_i - \sum_{i \neq j} W_{i,j} x_j||^2$. With the weight matrix W^*, we then search for a lower dimensional embedding $z_i \in \mathbb{R}^k$ where $k < p$ that minimizes $\sum_i ||z_i - \sum_{i \neq j} W^*_{i,j} z_j||^2$. Roweis and Saul (2000) called this method Locally Linear Embedding.

1.3 Sparse distance metric learning

Finding a suitable distance metric $\mathbf{M}$ involves filling out the $p \times p$ matrix $\mathbf{M}$. Therefore, it would be beneficial to consider *sparse* distance metric learning by looking at $\ell 1$-regularized metric learning. By enforcing sparsity, we perform variable selection and can eliminate the contribution of uninformative noise variables from the metric.

Sparse regression with $\ell 1$ penalization was originally proposed by Tibshirani (1996). He called this technique LASSO, which stands for "Least Absolute Shrinkage and Selection Operator" Fu (1998) considered the family of penalization $\sum |\beta_j|^\gamma$ for $\gamma \geq 1$. The penalty is not convex when $\gamma < 1$. The $\gamma > 1$ does not generally give sparse solution. $\ell 1$ penalization is interesting because it promotes a sparse solution while remaining a convex problem. Least angle regression (Efron et al., 2004) and the homotopy method (Osborne et al., 2000) provide a fast path following algorithm to solve a regularized squared error loss problem. Friedman et al. (2007) introduced the pathwise coordinate descent for other loss functions.

Sparse distance metric learning was first investigated by Rosales and Fung (2006). Rosales and Fung (2006) proposed adding a $\ell 1$ penalty to the entries of the matrix $\mathbf{M}$. This promotes the sparsity of the solution $\mathbf{M}$. They avoid the problem of dealing with the positive semi-definite constraint by imposing a more restrictive constraint that $\mathbf{M}$ should be diagonal dominant. A diagonal dominant matrix consists of entries such the sum of absolute value of off-diagonal entries in a row is smaller than or equal to the magnitude of the on-diagonal entries. A diagonal dominant matrix with non-negative diagonal entries is positive semi-definite by Gershgorin's Theorem. Therefore, the positive semidefinite condition of a diagonally dominant matrix is a set of linear constraints. This allows the problem to be solved by existing off-the-shelf

optimization algorithms. They further simplify the problem by penalizing only the diagonal entries of the matrix $\mathbf{M}$, with the off-diagonal entries implicitly penalized by the diagonal dominant constraint. The minimization of a loss function L with regularization becomes

$$\min_{\mathbf{M}:\mathbf{M}\succeq 0} L(\mathbf{M}) + \lambda \text{ tr}(\mathbf{M}) \text{ where } \mathbf{M} \text{ is diagonal dominant}$$

where $tr(\mathbf{M})$ denotes the trace of the matrix $\mathbf{M}$.

1.4 Our Contribution

Our contribution is to consider metric learning problem with penalization. The whole problem can be unified as one that minimizes a loss function L with a penalty term Ω.

$$\min_{\mathbf{M}:\mathbf{M}\succ 0} L(\mathbf{M}) + \Omega(\mathbf{M}).$$

Different choices of the set of variables, loss functions and penalties lead to different applications of the algorithm.

Three different $\ell 1$ penalties We study three different penalties Ω. The first penalty to be investigated is the entrywise lasso penalty where $\Omega(\mathbf{M}) = \|\mathbf{M}\|_1 = \sum_{i=1}^{p}\sum_{j=1}^{p}|\mathbf{M}_{i,j}|$ equals the sum of absolute values of the entries of $\mathbf{M}$. The lasso type penalty assumes sparsity on the individual entry of $\mathbf{M}$. The second penalty to be studied is the overlapping group lasso penalty $\Omega(\mathbf{M}) = \|\mathbf{M}\|_{group} = \sum_{i=1}^{p}\sum_{j=1}^{p}\sqrt{\mathbf{M}_{ij}^2}$. We demonstrate that the group penalty recovers $\mathbf{M}$ with zero rows/columns, equivalent to performing variable selection of the underlying covariates. Finally, we discuss the trace penalty $\Omega(\mathbf{M}) = tr(\mathbf{M})$ which equals the sum of the eigenvalues of $\mathbf{M}$. This penalty allows recovery of a low rank metric $\mathbf{M}$.

Consistency Theorem We prove consistency results for all three types of penalties based on compatibility condition (Bickel et al., 2009; Van De Geer and Bühlmann, 2009) when the loss function is squared error loss. Let $\mathbf{M}^*$ be the $p \times p$ true underlying metric and $\hat{\mathbf{M}}$ be the fitted metric. The consistency theories for all three type of penalties share the following form

$$\sum_{i>j}(d_{\mathbf{M}^*}(i,j) - d_{\hat{\mathbf{M}}}(i,j))^2/N + \lambda\Omega(\hat{\mathbf{M}} - \mathbf{M}^*) \leq 4\frac{\lambda^2}{\psi^2} \tag{1.8}$$

where the choice of the regularization parameter λ and the constant ψ differs for each type of regularization. The regularization parameter λ can be chosen to be of order $O(\sqrt{\log p/n})$, $O(\sqrt{p \log p/n})$ and $O(\sqrt{p^2 \log p/n})$ for lasso penalty, group lasso penalty and trace norm penalty respectively. Consistency theory of the form (1.8) gives an upper bound on the mean squared error of the fitted distances and a bound on the discrepancy between the fitted metric $\hat{\mathbf{M}}$ and the true metric $\mathbf{M}^*$. As fitting $\hat{\mathbf{M}}$ involves $O(p^2)$ parameters, both lasso penalty and group penalty provide an improved bound if sparsity assumption holds, while we believe that the result for the trace norm regularization could be sharpened further with stronger assumptions. The consistency theorem makes no guarantee that the fitted metric $\hat{\mathbf{M}}$ would be sparse, but the consistency theorem shows that the discrepancy between $\hat{\mathbf{M}}$ and $\mathbf{M}^*$ would be small

$$\Omega(\hat{\mathbf{M}} - \mathbf{M}^*) \leq 4\frac{\lambda}{\psi^2}.$$

Hence, the fitted metric $\hat{\mathbf{M}}$ would be approximately sparse if the underlying $\mathbf{M}^*$ is sparse.

Numerical Optimization We discuss two different algorithms in detail to perform distance metric learning. The first algorithm is the Alternating Direction Method of Multiplier (Boyd et al., 2011). For learning a diagonal $\mathbf{M}$, we discuss the pathwise coordinate descent scheme of Friedman et al. (2007).

Application The main application of distance metric learning is to learn a metric that improves the accuracy of nearest neighbour classification. We illustrate with examples that applying regularization allows us to achieve better accuracy and be more robust to noisy variables. The use of regularization also allows one to potentially consider a large number of covariates and we discuss how rule ensemble (Friedman and Popescu, 2008) can be used in the metric learning problem for classification. We conclude with an application in document search problem.

1.5 Summary of the chapters

The thesis is organized as follows.

Chapter 2 We consider squared error loss and show that sparse recovery properties hold for regularization $\Omega(\mathbf{M}) = ||\mathbf{M}||_1$, linking existing result of sparse recovery of LASSO with the metric learning problem. Its application in classification is shown via some numerical examples.

Chapter 3 We consider sparsity in the underlying variables: certain rows/columns of $\mathbf{M}$ are assumed to be zero. We demonstrate that it is possible to achieve such selection with an overlapping group lasso penalty with $\Omega(\mathbf{M}) = \|\mathbf{M}\|_{group} = \sum_{i=1}^{p} \sqrt{\sum_{j=1}^{p} \mathbf{M}_{ij}^2}$. We prove a consistency theory that applies to the group lasso and present an algorithm to solve the corresponding optimization problem.

Chapter 4 We explore another type of sparsity that $\mathbf{M}$ is a low rank matrix. We demonstrate that low rank recovery is possible using the trace norm regularization $\Omega(\mathbf{M}) = ||\mathbf{M}||_*$, which equals the sum of eigenvalues of $\mathbf{M}$. We show that there is a close connection between the trace norm penalty and the group lasso penalty in Chapter 3. A consistency theory for the trace norm penalty is proved. This chapter is concluded with numerical simulations comparing three forms of regularization.

Chapter 5 Imposing regularization allows one to potentially consider a very large number of covariates. In this chapter, we follow the proposal of Friedman and Popescu (2008) and expand the set of covariates to the set of rule ensembles. Rules are hyper-rectangles defined in the design space. They can be expressed as simple statements on the input variable values and can be interpreted easily.

Chapter 6 We illustrate an application of distance metric learning in a document retrieval application. Instead of explicit class labels information, the input required is that the user can decide whether document A is more similar to document B than to document C given three documents A, B and C. We propose a metric learning algorithm that aims to preserve this type of relative distances between pairs. The loss function is similar to the one proposed by Weinberger and Saul (2009) in Large Margin Nearest Neighbour. We apply this algorithm to a multi-label prediction problem.

Appendix A In this appendix, we include a short review of consistency of lasso and compatibility condition in a linear regression setting. We discuss compatibility condition (Van De Geer and Bühlmann, 2009) and its relationship with the smallest eigenvalue of the Gram matrix.

Chapter 2

Sparse distance metric learning

Data classification algorithm such as k-nearest neighbour can often benefit from learning a distance metric. A number of such metric learning algorithms have been proposed. Xing et al. (2002) suggested learning a metric such that the distances between observations of the same class are small and distances between observations of different classes are large. Goldberger et al. (2008) proposed neighbourhood component analysis which aims to maximize the prediction accuracy directly, resulting in a non-convex optimization problem. Weinberger and Saul (2009) introduced large margin nearest neighbour which tries to ensure the neighbourhood of an observation is surrounded by observations of the same class and relies on a pick of so-called target neighbours, whose closeness should be preserved in the transformation. Our contribution is to consider *sparse* distance metric learning by looking at $\ell 1$-regularized metric learning under squared error loss. By enforcing sparsity, we perform variable selection and can eliminate the contribution of uninformative noise variables from the metric. We first formulate our approach in Section 2.1 and prove a sparse recovery property in Section 2.2. Numerical examples are shown in Section 2.3.

2.1 Sparse Distance Metric Learning

Suppose there are n observations $X_1, \ldots, X_n \in \mathbb{R}^p$ with associated class labels $Y_i \in \{1, \ldots, K\}$. The squared distance between two observations i and j is defined under metric $\mathbf{M}$ as

$$d_{\mathbf{M}}(i,j) = (X_i - X_j)^T \mathbf{M} (X_i - X_j),$$

where $\mathbf{M}$ is a $p \times p$ positive semidefinite matrix. Under a Euclidean distance metric, $\mathbf{M}$ is a p-dimensional identity matrix. Since directly optimizing the accuracy of a nearest neighbour classifier is a non-smooth and non-convex problem, the usual approach is to propose an alternative convex optimization problem to improve the performance of nearest neighbour indirectly. Xing et al. (2002) proposed to learn an appropriate distance metric as

$$\hat{\mathbf{M}} \;=\; \text{argmin}_{\mathbf{M}:\mathbf{M}\succeq 0} \sum_{(i,j)\in G} d_{\mathbf{M}}(i,j) \quad \text{such that} \sum_{(i,j)\in D} d_{\mathbf{M}}(i,j)^{1/2} \geq 1, \quad (2.1)$$

where $\mathbf{M} \succeq 0$ means that $\mathbf{M}$ is positive semidefinite. Note that d refers to the squared distance in our approach, for simplicity of notation. The set G contains all pairs of observations of the same class $G = \{(i,j) : Y_i = Y_j\}$, while D contains all observations pairs of a different class. The proposed metric learning can successfully find a useful metric for many problems, yet it cannot recover a sparse solution for the metric which is interesting in higher-dimensional problems for better interpretability of the result. The proposals of Goldberger et al. (2008) and Weinberger and Saul (2009) likewise do not yield sparse results in general.

Adding a penalty to (2.1) might seem like an attractive way of achieving a sparse distance metric. However, the derivative of $d_{\mathbf{M}}(i,j)^{1/2}$ gets very large for observations in different classes with very small fitted distances. The algorithm is thus in practice very sensitive to those observation pairs of different classes with a small fitted distance and does not tend to perform well with a variable selection procedure. We follows thus a different approach by making a connection with linear regression and select for some $\alpha \geq 1$,

$$\hat{\mathbf{M}} \;=\; \text{argmin}_{\mathbf{M}:\mathbf{M}\succeq 0}\, L_\alpha(\mathbf{M}), \quad (2.2)$$

where, for $N = n(n-1)/2$,

$$L_\alpha(\mathbf{M}) = N^{-1} \sum_{i>j} |d(i,j) - d_{\mathbf{M}}(i,j)|^\alpha \quad (2.3)$$

for a target distance $d(i,j)$. Here, we use the simplest approach and set $d(i,j) = 1\{Y_i \neq Y_j\}$. The constant 1 is arbitrary, just as in (2.1), and can be replaced by a different constant. The estimator under (2.3) is not yet sparse, that is all elements of $\mathbf{M}$ will in general have a non-zero value. To increase interpretability and predictive

accuracy in the presence of many noise variables, we can add a sparsity penalty to get

$$\hat{\mathbf{M}}(\lambda) \quad = \quad \text{argmin}_{\mathbf{M}:\mathbf{M}\succeq 0}\, L(\mathbf{M}) + \lambda\|\mathbf{M}\|_1, \tag{2.4}$$

where the penalty $\|\mathbf{M}\|_1 = \sum_{i=1}^{p}\sum_{j=1}^{p}|M_{i,j}|$ is the entrywise $\ell 1$ norm of the matrix $\mathbf{M}$. We penalize both the diagonal and off-diagonal elements of $\mathbf{M}$ as we also want to reduce the number of non-zero diagonal elements in $\mathbf{M}$ in many applications. We also examine the effectiveness of diagonal sparse distance metric learning,

$$\hat{\mathbf{M}}(\lambda) \quad = \quad \text{argmin}_{\mathbf{M}:\mathbf{M} \text{ is diagonal } \mathbf{M}\succeq 0}\, L(\mathbf{M}) + \lambda\|\mathbf{M}\|_1. \tag{2.5}$$

The diagonal distance metric learning (2.5) clearly has the advantage that the solution is much faster to obtain computationally and the result is easier to interpret since the coefficients corresponds to a re-weighting of the variables and zero entries remove a variable completely from the final distance metric. We might lose a bit on predictive accuracy, however, and these tradeoffs will be examined later. For a simpler notation, we shall use the notation $\hat{\mathbf{M}}$ to denote $\hat{\mathbf{M}}(\lambda)$ for the rest of this paper and the reader should be aware that $\hat{\mathbf{M}}$ is dependent on λ.

2.2 Sparse Recovery

We will prove a sparse recovery result for an optimal sparse metric. We first state some necessary assumptions. We will work with a value $\alpha = 2$. While $\alpha = 1$ in (2.3) might seem more natural since it penalizes large discrepancies less heavily, results tend to be similar in practice and $\alpha = 2$ is computationally more convenient since the loss function will be quadratic in the entries of $\mathbf{M}$ and allows us to leverage insights from sparse high-dimensional regression to sparse distance metric learning.

2.2.1 Assumptions

If the best approximating matrix $\mathbf{M}^*$ is sparse, we will show that we can recover it using the following $\ell 1$-regularized problem. We look at loss function (2.3). Let the

best approximating metric for fixed observations $X_1, \ldots, X_n$ be given by $\mathbf{M}^*$,

$$\begin{aligned}\mathbf{M}^* &= \text{argmin}_{\mathbf{M}:\mathbf{M}\succeq 0}\, E\big[L(\mathbf{M})\big] \\ &= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\big[P(Y_i \neq Y_j)|1 - d_{\mathbf{M}}(i,j)|^{\alpha} + P(Y_i = Y_j)|0 - d_{\mathbf{M}}(i,j)|^{\alpha}\big],\end{aligned}$$

where the expectation is with respect to random class labels $Y_1, \ldots, Y_n$. Assume that $\mathbf{M}^*$ is unique. Let also $d^*(i,j) = (X_i - X_j)^T\mathbf{M}^*(X_i - X_j)$ be the squared distance under an optimal metric and let $\varepsilon_{i,j} = d^*(i,j) - d(i,j) = d^*(i,j) - 1\{Y_i \neq Y_j\}$ be the difference between the observed and actual squared distances. We make two assumptions on the error.

Assumption 2.2.1 *We assume that the error is bounded, that is $|\varepsilon_{i,j}| \leq \delta^2$, and has mean zero.*

This boundedness assumption is trivially fulfilled for classification tasks as long as the distances under the optimal metric are bounded. One could in practice fit an intercept term so that $\varepsilon_{i,j}$ has mean zero.

Since we define $d(i,j)$ using the underlying Y_i, errors are correlated with the underlying observation pairs. The distance $d(i,j)$ is for example correlated with the distance $d(i,k)$. The set of all $\varepsilon_{i,j}$ would therefore not be independent.

Assumption 2.2.2 *The errors $\varepsilon_{i,j}$ and $\varepsilon_{h,k}$ are independent if indices i,j,h,k do not overlap.*

Bian and Tao (2011) gave a proof to show that the set of all pairwise distances is dominated by independent pairs, therefore, asymptotically, the empirical loss function converges to the expected loss at the rate of $\mathrm{O}(n^{-1/2})$. Their result used the second moment to derive the necessary inequality and the rate of convergence given by their result is not fast enough to consider the case when the number of variables exceeds the number of observations. We present a different proof which gives an exponential tail bound which could then be applied to the $\ell 1$-regularized variable selection procedure.

2.2.2 Consistency Results

This section follows closely the current results of Lasso in least squared regression. We have to assume some conditions on the design matrix in order to proceed with our proof. Intuitively, we would require the design matrix to be not too correlated.

We consider a version of the compatibility condition/restricted eigenvalue assumption used in Bickel et al. (2009). Let S be the set of indices that $\mathbf{M}^*$ is non-zero,

$$S = \{(h,k) : \mathbf{M}^*_{h,k} \neq 0\} \text{ where } h = 1,\ldots,p \text{ and } k = 1,\ldots,p,$$

and let S^c denotes the complement of S. Let $\mathbf{A}$ be a $p \times p$ matrix. Let $\mathbf{A}_S$ denotes the matrix with entries $\mathbf{A}_{S(h,k)} = \mathbf{A}_{h,k}1((h,k) \in S)$. We can then state a compatibility condition analogous to the regression variant in Bühlmann and van de Geer (2011).

Assumption 2.2.3 (Compatibility Condition) *There exists $\psi > 0$ such that for any symmetric matrix $\mathbf{A}$ with $\|\mathbf{A}_{S^c}\|_1 \leq 3\|\mathbf{A}_S\|_1$,*

$$\|\mathbf{A}_S\|_1^2 \leq \frac{1}{N\psi^2} \sum_{i>j} tr\big[\mathbf{A}(X_i - X_j)(X_i - X_j)^T\big]^2$$

The trace operation defines a dot product in the symmetric matrix space. Therefore, if $(X_i - X_j)(X_i - X_j)^T$, $i,j \in \{1,\ldots,n\}$ spans the space of symmetric matrix of size $p \times p$, then at least one term of $\mathrm{tr}\big[\mathbf{A}(X_i - X_j)(X_i - X_j)^T\big]$ must be non-zero for any $\mathbf{A}$ and hence compatibility condition holds. When the number of observations exceed the number of parameters, then $(X_i - X_j)(X_i - X_j)^T$, $i,j \in \{1,\ldots,n\}$ cannot span the space of symmetric matrix and the set $E = \{\mathbf{A}$ where $\sum_{i>j} \mathrm{tr}\big[\mathbf{A}(X_i - X_j)(X_i - X_j)^T\big]^2 = 0\}$ is non-empty. The restriction $\|\mathbf{A}_{S^c}\|_1 \leq 3\|\mathbf{A}_S\|_1$ allows us to obtain a positive ψ when the set E does not intersect with the cone $\|\mathbf{A}_{S^c}\|_1 \leq 3\|\mathbf{A}_S\|_1$. We note that it is common to have a term of the cardinality of non-zero entries on the right hand side of the inequality, we scale the ψ accordingly to absorb that term for a simpler notation. More discussion of the compatibility condition and its connection with other conditions can be found in Van De Geer and Bühlmann (2009).

We are now ready to state our theorem.

Theorem 2.2.4 *Let $\hat{\mathbf{M}}$ be the estimator in (2.4) with loss function (2.3) and $\alpha = 2$. Suppose that Assumptions 2.2.2 and 2.2.1 hold. Suppose that there exists some $c > 0$ such that $|X_{i,h}| \leq c$ for all $h = 1,\ldots,p$, and $i = 1,\ldots,n$. Let*

$$\lambda := 16c^2\delta^2\sqrt{\frac{t^2 + 4\log(p) + 2\log(n)}{(n-1)/2}}.$$

Then, with probability at least $1 - 2\exp(-t^2/2)$,

$$N^{-1}\sum_{ij}\big[d_{\hat{\mathbf{M}}}(i,j) - d^*(i,j)\big]^2 \;+\; \lambda\|\hat{\mathbf{M}} - \mathbf{M}^*\|_1 \;\leq\; \frac{4\lambda^2}{\psi^2}.$$

A proof is given in section 2.5. It leverages existing related results for high-dimensional regression. We thus obtain a bound on the estimation error and of the fitted distances and the $\ell 1$-error of the found sparse metric. It implies (leaving aside logarithmic factors) a $n^{-1/2}$-scaling for the $\ell 1$-distance between the fitted metric $\hat{\mathbf{M}}$ and the best approximating sparse metric $\mathbf{M}^*$. A well-approximating distance metric can thus be found in a high-dimensional setup in the presence of many noise variables under the condition that the best approximating distance metric is sparse.

2.3 Numerical example

We look at some numerical examples for estimator (2.3) under a fixed value of α. To compute the estimator (2.3) for a fixed value of λ and $\alpha = 2$, we apply the gradient projection algorithm presented in Bertsekas (1999) in Section 2.3. Similar algorithms are also used in Xing et al. (2002) and Weinberger and Saul (2009). We first perform a gradient descent and then project the proposed point back to the positive semi-definite cone. We compute a sequence of optimization with a decreasing regularization parameter λ as suggested in Friedman et al. (2009). The solution with larger regularization gives a warm starting point for the less regularized problem. By managing the non-zero entries of the problem, the active indices are often of a much lower dimension and therefore easier to compute. The test accuracy is obtained by performing nearest neighbour using the learned metric.

As a comparison, we perform greedy forward selection and random forest (Breiman, 2001). The forward selection algorithm begins with a zero metric $\mathbf{M} = \mathbf{0}$. Addition of variable is performed by setting the corresponding diagonal entry to 1. The variable that minimizes nearest neighbour error is added to the model at each step. The above process is repeated until the cross validated error cannot improve further. The fitted metric $\hat{\mathbf{M}}$ is a diagonal matrix with entries either 0 or 1. This forward selection method has the benefit of directly optimizing the nearest neighbour accuracy.

2.3.1 European Voting dataset

This dataset consists of the decisions made by the members of the European Parliament in the roll call votes (Hix et al., 2006). Our aim is to investigate which are the

important decisions that separate different European Parliament Groups.

We focus on the period between 1996-2000, removing members of parliament who are absent in more than 1500 votes and simplify the decision taken by parliament members as either one of a positive response or a negative one. Voting "no", "abstain", "present but did not vote" and "absent" all count as a negative response.

We divide the dataset into two partitions: a test set of size 110 and a training set of size 419. The amount of regularization is chosen by cross validation. Since we are interested in the importance of individual votes and want to ignore the interactions between votes for now, we learned a diagonal metric $\hat{\mathbf{M}}$ by forcing all off-diagonal elements to be 0. We repeat the simulation 10 times. Out of the 3740 roll call votes, our algorithm has chosen on average 222.1 non-zero coefficients (with a standard deviation of 28.1).

The average predictive performance is shown below as the percentage of correctly predicted classes for the test data observations. The values in brackets are the associated standard deviation.

Metric learning($\mathbf{M} = \hat{\mathbf{M}}$)	Euclidean ($\mathbf{M} = \mathbf{I}$)	Random Forest	Forward Selection
96.9(1.1)	93.8(1.7)	97.6(1.1)	92.3(2.0)

Metric learning with a diagonal $\hat{\mathbf{M}}$ is in this example equally powerful as Random Forests (Breiman, 2001), although nearest neighbour under the Euclidean metric is also similarly powerful, suggesting that equal-weighting of votes provides a good approximation to a metric that allows one to distinguish between different fractions in parliament. The greedy selection, however, performs slightly worse than other methods.

As an illustration of variable selection effectiveness, we need to have information that some votes are un-informative. To this end, we permute the decisions on 90% of the roll call votes. Out of the 3740 roll call votes, our algorithm has now chosen on average 95.3 non-zero coefficients with a standard deviation of 20.7. On average, 11.5 of the chosen votes belong to the permuted votes with a standard deviation of 14.2. The predictive accuracy is now:

Metric learning($\mathbf{M} = \hat{\mathbf{M}}$)	Euclidean ($\mathbf{M} = \mathbf{I}$)	Random Forest	Forward Selection
95.0(2.6)	83.0(2.7)	88.2(3.1)	86.6(3.2)

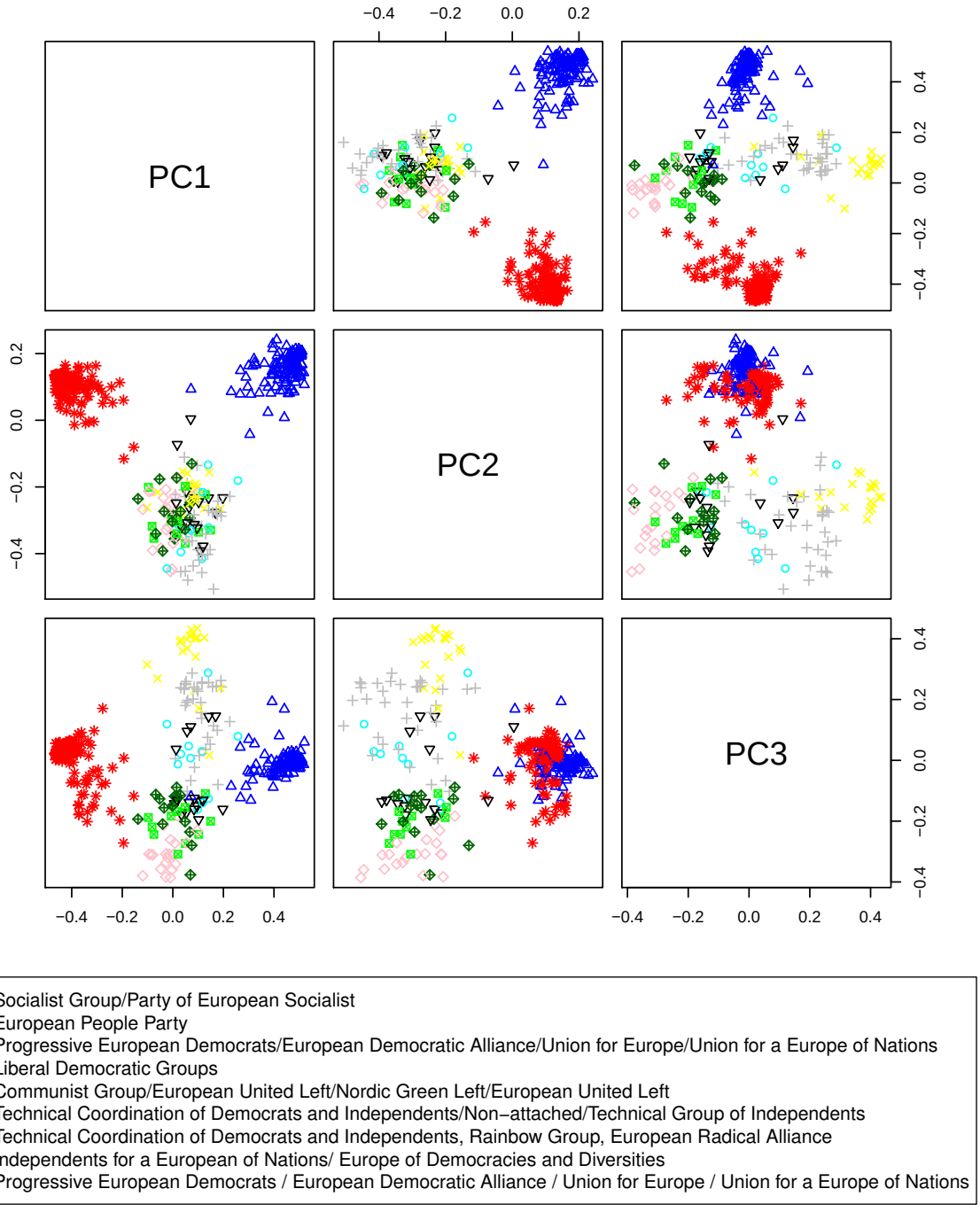


Figure 2.1: A PCA projection of the European vote dataset after learning a diagonal metric. The two biggest parliamentary fractions are Socialist Group/Party of European Socialist (red $*$), European People Party (blue triangle).

The metric learning approach has suffered the least in predictive accuracy when permuting some of the votes and thus increasing the amount of noise variables. The Nearest Neighbour algorithm under the original Euclidean metric, in contrast, has suffered a steep decline in predictive accuracy as most of the votes are now uninformative but contribute strongly to the Euclidean distance. The forward selection also shows a steep decline in predictive accuracy.

To visualize the result, we can perform an eigendecomposition of $\hat{\mathbf{M}} = \mathbf{V}^T\mathbf{D}\mathbf{V}$, where $\mathbf{D}$ is a diagonal matrix of eigenvalues and $\mathbf{V}$ consists of the corresponding eigenvectors. We define $\mathbf{C} = \mathbf{D}^{\frac{1}{2}}\mathbf{V}$, where $\mathbf{D}^{\frac{1}{2}}$ is a diagonal matrix with the square root of the entries of $\mathbf{D}$. We can then rewrite the distances as

$$d_{\hat{\mathbf{M}}}(i,j) = (\mathbf{C}X_i - \mathbf{C}X_j)^T(\mathbf{C}X_i - \mathbf{C}X_j). \tag{2.6}$$

Using $\mathbf{C}$ as a transformation for the data, the estimated distance between two points is simply the Euclidean distance after the transformation. A visualization of the first few eigenvectors is shown in Figure 2.1, showing a reasonable separation between parliamentary groups.

2.3.2 Other datasets

More datasets from the UCI Machine Learning Repository (Frank and Asuncion, 2010) are investigated. Categorical variables are converted into indicator variables.

	Number of observations	Number of variables	Number of classes
Vehicle	846	18	4
Vowel	990	24	11
Soybean	562	65	19
Zoo	101	16	7

We randomly split the data into a training set (80%) and a testing set (20%) and we repeat the simulations 10 times. The amount of regularization is determined by cross validation. We also compare the accuracy with random forest (Breiman, 2001), another metric learning algorithm Large Margin Nearest Neighbour (LMNN) (Weinberger and Saul, 2009) and Large Margin Nearest Neighbour restricted to a diagonal metric. The following is the prediction accuracy of the test set. (The values in brackets are the associated standard deviations.)

	Vowel	Zoo	Vehicle	Soybean
Metric Learning($\mathbf{M} = \hat{\mathbf{M}}$)	96.3 (2.8)	92.4 (4.6)	58.5 (5.1)	90.3 (3.0)
1-NN ($\mathbf{M} = \mathbf{I}$)	93.1 (2.8)	92.4 (6.4)	65.2 (3.2)	92.9 (2.4)
Random Forest	96.6 (1.5)	95.7 (4.2)	76.8 (3.8)	86.4 (2.6)
LMNN	96.6 (1.1)	93.3 (6.0)	76.9 (3.9)	90.6 (3.3)
LMNN(diagonal)	95.3 (2.2)	91.4 (8.3)	71.0 (4.4)	88.9 (3.0)
Forward Selection	98.2 (1.0)	92.8 (6.4)	69.1 (2.8)	89.2 (4.1)

Performing nearest neighbour on the original dataset already yields very good performance. Our method gives very similar predictive performance. To investigate the variable selection properties, we generate 100 additional noise variables by adding randomly permuted existing variables to the dataset.

	Vowel	Zoo	Vehicle	Soybean
Metric Learning($\mathbf{M} = \hat{\mathbf{M}}$)	88.3 (3.1)	94.0 (3.9)	58.4 (3.8)	91.3 (2.7)
1-NN ($\mathbf{M} = \mathbf{I}$)	25.7 (1.5)	75.8 (7.0)	39.2 (3.9)	73.9 (4.0)
Random Forest	81.3 (2.8)	93.0 (7.9)	71.1 (3.3)	91.4 (2.2)
LMNN	59.9 (5.3)	86.7 (8.5)	44.1 (11.4)	91.2 (1.7)
LMNN(diagonal)	43.8 (5.3)	92.9 (6.7)	49.8 (2.6)	89.1 (2.5)
Forward Selection	98.1 (1.0)	91.9 (6.8)	60.2 (3.7)	87.9 (2.6)

As expected, the performance of nearest neighbour under a Euclidean metric drops drastically if there are additional noise variables. By learning a suitable metric, we can still maintain a reasonable performance. Forward selection performs quite well with Vowel data, which suggest that there be an advantage by directly optimizing nearest neighbour accuracy.

Metric learning fits a sparse distance metric and it is interesting to see how many of the selected variables (that is non-zero terms on the diagonal of $\hat{\mathbf{M}}$) belong to the original unpermuted set or the noise variables on average. The following table reports the average number of variables selected in the simulations.

	Original variables	Noise variables
Vehicle	3.5 (0.53)	2.0 (1.76)
Vowel	5.2 (1.14)	1.5 (1.43)
Soybean	42.3 (6.77)	16.1 (14.59)
Zoo	6.4 (2.07)	0.2 (0.63)

Very few noise variables are chosen in the selected models.

So far, we have been trying to learn a diagonal $\hat{\mathbf{M}}$ according to (2.5). When learning a full distance metric as in (2.4), the predictive accuracy does not change appreciably. The percentage of correctly classified test samples is 96.1 for the Zoo, 57.4 for the Vehicle, 90.9 for the Soybean dataset and 92.2 for the Vowel dataset. Given the higher

computational cost of the full metric learning approach, it seems as if restricting $\hat{\mathbf{M}}$ to be diagonal as in (2.5) is preferable in practice for many examples.

We have thus shown at hand of a few examples that distance metric learning has the attractive property of having very good predictive accuracy in the presence of many noise variables and, additionally, has a variable selection property, which is in contrast to the distance metric learning approaches in Xing et al. (2002) and Weinberger and Saul (2009). Random Forests also generates a variable importance measure, but would in general not exclude un-important variables completely from the fitted trees.

2.3.3 Zoo dataset

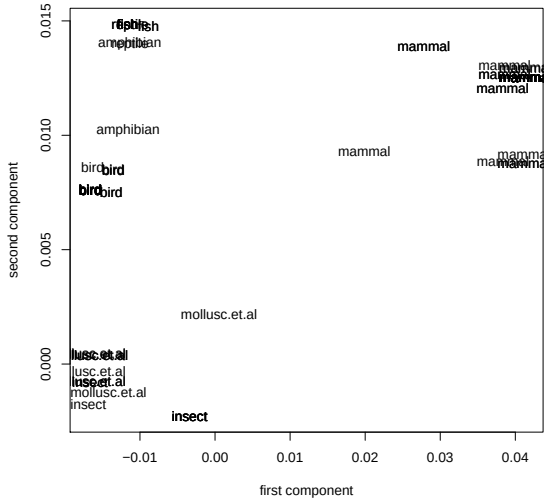
As an illustration of how one could interpret a full metric $\mathbf{M}$, we return to the Zoo dataset from the UCI respository. It consists of 101 animals, each of which is classified as one of mammal, bird, reptile, fish, amphibian, insect, mollusc et al. We are given 16 attributes of each animal: hair, feathers, eggs, milk, airborne, aquatic, predator, toothed, backbone, breathes, venomous, fins, legs, tail, domestic, catsize. Legs have numeric value and others are binary variables. The problem is to predict the classification given above 16 attributes. We perform metric learning of the full matrix $\mathbf{M}$ and the amount of regularization is chosen by cross validation.

We try to interpret $\mathbf{M}$ by performing eigendecomposition:

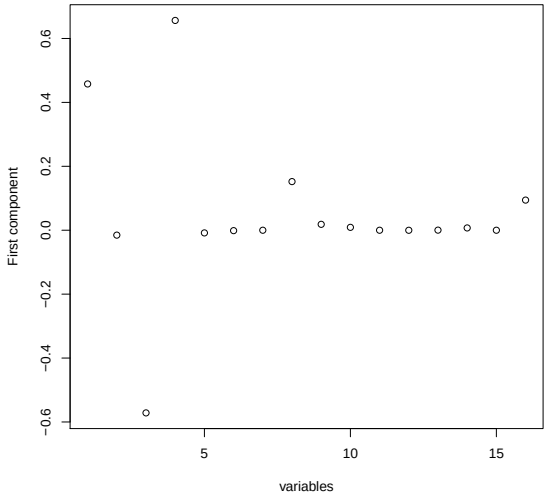
$$\mathbf{M} = \mathbf{V}^T\mathbf{D}\mathbf{V}$$

where $\mathbf{V}$ is a matrix of eigenvectors and $\mathbf{D}$ is a diagonal matrix of eigenvalues. The eigenvectors associated with the top two eigenvalues provides a rank-2 transformation matrix. This 2D-embedding is optimal in the sense that this is the best rank 2 approximation of $\mathbf{M}$ under Frobenius norm.

We first project the data onto the first two eigenvectors of $\mathbf{M}$. The projection given by first two eigenvectors shows interesting separation of the animals.

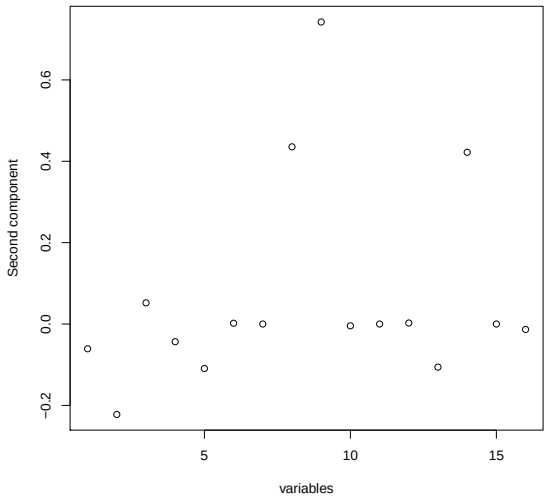


The first component separates mammals from other type of animals. It would be interesting to investigate the coefficients of the first eigenvectors.



The three largest coefficients suggest that an animal with hair, milk but no eggs would be a mammal.

We then look at the second eigenvectors of $\mathbf{M}$.



The largest coefficients in descending order are backbone, toothed and tail. We can see that insects are separated from other animals in the second component due to a lack of backbone.

2.4 Summary and Discussion

We have proposed a way to learn sparse metrics for multiclass classification problems and shown that it is possible to recover a sparse metric in the presence of many noise variables. The theoretical results build upon similar conditions to the regression case. Numerical results showed the resilience of the method when faced with many noise variables. The predictive performance is very competitive and does not suffer as much as other methods when the number of noise variables is increased.

2.5 Proofs

In this section, we present the proofs of the results we stated in Section 2.2.2.

2.5.1 Additional Lemmata

Lemma 2.5.1 *Suppose assumptions 2.2.1 and 2.2.2 hold and suppose the covariates are bounded, that is there exists some* $c > 0$ *such that* $|X_{i,h}| \leq c$ *for all* $h = 1, \ldots, p$,

and $i = 1, \ldots, n$. *Let*

$$\lambda_0 := 8c^2\delta^2\sqrt{\frac{t^2 + 4\log(p) + 2\log(n)}{(n-1)/2}}.$$

Then, for $N = n(n-1)/2$,

$$\mathbf{P}\left(\max_{1\leq h\leq p, 1\leq \ell\leq p} |2N^{-1}\sum_{ij}\varepsilon_{i,j}(X_{ih} - X_{jh})(X_{i\ell} - X_{j\ell})| < \lambda_0\right) \geq 1 - 2\exp\left(-\frac{t^2}{2}\right).$$

Proof The main difficulty here is the fact that the pairwise distances are not independent. When the number of observations n is odd, we can pick a set of $(n-1)/2$ pairwise distances so that no indices overlap and hence are independent. For this set of pairwise distances, we can show that the contribution from the noise term can be bounded with high probability. We can then show that the contribution of noise to the set of all pairwise distances can also be bounded by applying an union bound.

We first consider the case when n is odd. We will use a result from graph theory to prove our claim. A complete graph with n vertices is a graph where there is an edge between any two vertices. A pairwise distance can be represented as an edge in the complete graph. An edge colouring of a graph is that we can assign a colour to each edge such that no adjacent edges share the same colour. This implies that pairwise distances corresponding to edges of the same colour are independent under assumption 2.2.2. Edge chromatic number is the minimum number of colours required to colour the edges of a graph. A complete graph with n vertices has an edge chromatic number of n when n is odd and $n-1$ when n is even (a constructive proof is given in Soifer et al. (2008, P.133)), from which we can deduce that there is a colouring of the edges such that each colour consists of exactly $(n-1)/2$ edges when n is odd. Therefore, the set of all pairwise distances can be partitioned into n partitions $G_1, \ldots, G_n$, each partition consisting of $(n-1)/2$ pairwise distances such that no indices overlap. This partition is always possible because there exists a colouring of the edges with n colours in a complete graph which would give such a partition, with the edges of each colour forming a partition.

Let $V_{h,\ell,G_k} = \frac{2}{n-1}\sum_{\{i,j\}\in G_k} 2\varepsilon_{i,j}(X_{ih} - X_{jh})(X_{i\ell} - X_{j\ell})$.

By the boundedness of X and ε,

$$-8c^2\delta^2 \leq 2\varepsilon_{i,j}(X_{ih} - X_{jh})(X_{i\ell} - X_{j\ell}) \leq 8c^2\delta^2.$$

Therefore, by Hoeffding's inequality,

$$P(|V_{h,\ell,G_k}| \geq \lambda_0) \leq 2\exp\left[-\frac{2\lambda_0^2\left(\frac{n-1}{2}\right)^2}{\frac{n-1}{2}(16c^2\delta^2)^2}\right]$$
$$\leq 2\exp(-\frac{t^2+2\log n+4\log p}{2}).$$

By a union bound over all choice of h and ℓ,

$$P(\max_{1\leq h\leq p,1\leq \ell\leq p}|V_{h,\ell,G_k}| \geq \lambda_0) \leq 2p^2\exp(-\frac{t^2+4\log p+2\log n}{2})$$
$$= 2\exp(-\frac{t^2+2\log n}{2}).$$

Finally, by a simple union bound over all partitions of G_k,

$$P(\max_k \max_{1\leq h\leq p,1\leq \ell\leq p}|V_{h,\ell,G_k}| \geq \lambda_0) \leq 2n\exp(-\frac{t^2+2\log n}{2})$$
$$= 2\exp(-\frac{t^2}{2}).$$

And thus

$$\begin{aligned}
& \mathbf{P}(\max_{1\leq h\leq p,1\leq \ell\leq p}|2N^{-1}\sum_{ij}\varepsilon_{i,j}(X_{ih}-X_{jh})(X_{i\ell}-X_{j\ell})| < \lambda_0) \\
= \ & \mathbf{P}(\max_{1\leq h\leq p,1\leq \ell\leq p} n^{-1}|\sum_k V_{h,\ell,G_k}| < \lambda_0) \\
\geq \ & \mathbf{P}(\max_k \max_{1\leq h\leq p,1\leq \ell\leq p}|V_{h,\ell,G_k}| < \lambda_0) \\
\geq \ & 1-2\exp\big(-t^2/2\big).
\end{aligned}$$

When n is even, we can decompose the complete graph into $n-1$ sets of disjoint pairs, each set with n pairwise distances. We can derive a slightly stronger bound in this case than the case when n is odd. ∎

The proofs of Lemmata 2.5.2 and 2.5.4 follow closely the ones given in Chapter 6 of Bühlmann and van de Geer (2011), with modifications to handle the matrix notations.

Lemma 2.5.2 *Assume that* $\max_{1\leq h\leq p,1\leq \ell\leq p}|2N^{-1}\sum_{ij}\varepsilon_{i,j}(X_{ih}-X_{jh})(X_{i\ell}-X_{j\ell})| < \lambda_0$. *Then*

$$N^{-1}\sum_{ij}(\hat{d}(i,j)-d^*(i,j))^2+\lambda\|\hat{\mathbf{M}}\|_1 \leq \lambda_0\|\hat{\mathbf{M}}-\mathbf{M}^*\|_1+\lambda\|\mathbf{M}^*\|_1.$$

Proof Since $\hat{\mathbf{M}}$ minimizes $L(\mathbf{M})$, we have $L(\hat{\mathbf{M}}) \leq L(\mathbf{M}^*)$ and thus

$$\begin{aligned}
\sum_{ij} \frac{(d(i,j) - \hat{d}(i,j))^2}{N} + \lambda\|\hat{\mathbf{M}}\|_1 &\leq \sum_{ij} \frac{(d(i,j) - d^*(i,j))^2}{N} + \lambda\|\mathbf{M}^*\|_1 \\
-2\sum_{ij} \frac{\hat{d}(i,j)d(i,j)}{N} + \sum_{ij} \frac{\hat{d}(i,j)^2}{N} + \lambda\|\hat{\mathbf{M}}\|_1 &\leq -2\sum_{ij} \frac{d^*(i,j)d(i,j)}{N} + \sum_{ij} \frac{d^*(i,j)}{N}\lambda\|\mathbf{M}^*\|_1 \\
\sum_{ij} \frac{(\hat{d}(i,j) - d^*(i,j))^2}{N} + \lambda\|\hat{\mathbf{M}}\|_1 &\leq \sum_{ij} 2(d(i,j) - d^*(i,j))\frac{(\hat{d}(i,j) - d^*(i,j))}{N} + \lambda\|\mathbf{M}^*\|_1 \\
\sum_{ij} \frac{(\hat{d}(i,j) - d^*(i,j))^2}{N} + \lambda\|\hat{\mathbf{M}}\|_1 &\leq \sum_{ij} 2\varepsilon_{i,j}\frac{(\hat{d}(i,j) - d^*(i,j))}{N} + \lambda\|\mathbf{M}^*\|_1 \\
&= \sum_{ij} 2\varepsilon_{i,j} tr((\hat{\mathbf{M}} - \mathbf{M}^*)(X_i - X_j)(X_i - X_j)^T)/N \\
&\quad + \lambda\|\mathbf{M}^*\|_1 \\
&= tr((\hat{\mathbf{M}} - \mathbf{M}^*)\sum_{ij} 2\varepsilon_{i,j}(X_i - X_j)(X_i - X_j)^T)/N \\
&\quad + \lambda\|\mathbf{M}^*\|_1 \\
&\leq \lambda_0\|\hat{\mathbf{M}} - \mathbf{M}^*\|_1 + \lambda\|\mathbf{M}^*\|_1,
\end{aligned}$$

which completes the proof. ∎

The following lemma shows that the compatibility condition only needs to hold over a cone $3\|\mathbf{M}_S\|_1 \geq \|\mathbf{M}_{S_c}\|_1$.

Lemma 2.5.3 *Assume that* $N^{-1}\max_{1\leq h\leq p, 1\leq \ell\leq p} |2\sum_{ij} \varepsilon_{i,j}(X_{ih} - X_{jh})(X_{i\ell} - X_{j\ell})| < \lambda_0$ *holds. Let* $\Delta = \hat{\mathbf{M}} - \mathbf{M}^*$. *If* $\lambda \geq 2\lambda_0$, *then* $3\|\Delta_S\|_1 \geq \|\Delta_{S_c}\|_1$.

First, note that

Proof

$$\begin{aligned}
\|\hat{\mathbf{M}}\|_1 &= \|\hat{\mathbf{M}} - \mathbf{M}^* + \mathbf{M}^*\|_1 \\
&= \|\Delta + \mathbf{M}^*\|_1 \\
&\geq \|\mathbf{M}^* + \Delta_{S_c}\|_1 - \|\Delta_S\|_1 \\
&= \|\mathbf{M}^*\|_1 + \|\Delta_{S_c}\|_1 - \|\Delta_S\|_1.
\end{aligned}$$

Then, by Lemma 2.5.2 and the choice that $\lambda \geq 2\lambda_0$,

$$\begin{aligned}
2\lambda\|\hat{\mathbf{M}}\|_1 &\leq \lambda\|\hat{\mathbf{M}} - \mathbf{M}^*\|_1 + 2\lambda\|\mathbf{M}^*\|_1 \\
2(\|\mathbf{M}^*\|_1 + \|\Delta_{S_c}\|_1 - \|\Delta_S\|_1) &\leq \|\Delta\|_1 + 2\|\mathbf{M}^*\|_1 \\
2\|\Delta_{S_c}\|_1 - 2\|\Delta_S\|_1 &\leq \|\Delta_S\|_1 + \|\Delta_{S_c}\|_1 \\
\|\Delta_{S_c}\|_1 &\leq 3\|\Delta_S\|_1.
\end{aligned}$$

The following lemma is a technical step to prove Theorem 2.2.4.

Lemma 2.5.4 *Assume that* $N^{-1}\max_{1\leq h\leq p, 1\leq \ell\leq p}|2\sum_{ij}\varepsilon_{i,j}(X_{ih} - X_{jh})(X_{i\ell} - X_{j\ell})| < \lambda_0$ *holds. By picking* $\lambda \geq 2\lambda_0$*,*

$$2\sum_{ij}\frac{(\hat{d}(i,j) - d^*(i,j))^2}{N} + \lambda\|\hat{\mathbf{M}}_{S_c}\|_1 \leq 3\lambda\|\hat{\mathbf{M}}_S - \mathbf{M}^*_S\|_1.$$

Proof By triangle inequality,

$$\|\hat{\mathbf{M}}\| \geq \|\mathbf{M}^*_S\| - \|\hat{\mathbf{M}}_S - \mathbf{M}^*_S\| + \|\hat{\mathbf{M}}_{S^c}\|.$$

Also note that we can expand $\|\hat{\mathbf{M}} - \mathbf{M}^*\| = \|\hat{\mathbf{M}}_S - \mathbf{M}^*_S\| + \|\hat{\mathbf{M}}_{S^c}\|$. We can hence further extend the result of Lemma 2.5.2,

$$\begin{aligned}
2\sum_{ij}\frac{(\hat{d}(i,j) - d^*(i,j))^2}{N} + 2\lambda\|\hat{\mathbf{M}}\|_1 &\leq 2\lambda_0\|\hat{\mathbf{M}} - \mathbf{M}^*\|_1 + 2\lambda\|\mathbf{M}^*\|_1 \\
2\sum_{ij}\frac{(\hat{d}(i,j) - d^*(i,j))^2}{N} + 2\lambda(\|\mathbf{M}^*_S\|_1 - \|\hat{\mathbf{M}}_S - \mathbf{M}^*_S\|_1 + \|\hat{\mathbf{M}}_{S^c}\|_1) &\leq \lambda\|\hat{\mathbf{M}} - \mathbf{M}^*\|_1 + 2\lambda\|\mathbf{M}^*\|_1 \\
2\sum_{ij}\frac{(\hat{d}(i,j) - d^*(i,j))^2}{N} + 2\lambda(\|\mathbf{M}^*_S\|_1 - \|\hat{\mathbf{M}}_S - \mathbf{M}^*_S\|_1 + \|\hat{\mathbf{M}}_{S^c}\|_1) &\leq \lambda(\|\hat{\mathbf{M}}_S - \mathbf{M}^*_S\|_1 + \|\hat{\mathbf{M}}_{S_c}\|_1) \\
&\quad + 2\lambda\|\mathbf{M}^*\|_1 \\
2\sum_{ij}\frac{(\hat{d}(i,j) - d^*(i,j))^2}{N} + \lambda\|\hat{\mathbf{M}}_{S^c}\|_1 &\leq 3\lambda\|\hat{\mathbf{M}}_S - \mathbf{M}^*_S\|_1 - \lambda\|\hat{\mathbf{M}}_{S_c}\|_1 \\
2\sum_{ij}\frac{(\hat{d}(i,j) - d^*(i,j))^2}{N} + \lambda\|\hat{\mathbf{M}}_{S_c}\|_1 &\leq 3\lambda\|\hat{\mathbf{M}}_S - \mathbf{M}^*_S\|_1,
\end{aligned}$$

which completes the proof. ∎

2.5.2 Proof of Theorem 2.2.4

Finally, we can prove the performance of the regularized method. By Lemma 2.5.1, it holds with probability at least $1 - 2\exp(-t^2/2)$ that

$$\max_{1\leq h\leq p, 1\leq \ell\leq p} |2N^{-1}\sum_{ij} \varepsilon_{i,j}(X_{ih} - X_{jh})(X_{i\ell} - X_{j\ell})| < \frac{\lambda}{2}.$$

Hence, using Lemma 2.5.4, it holds with probability $1 - 2\exp(-t^2/2)$ that

$$2\sum_{ij} \frac{(\hat{d}(i,j) - d^*(i,j))^2}{N} + \lambda\|\hat{\mathbf{M}} - \mathbf{M}^*\|_1 \ \leq \ 4\lambda\|\hat{\mathbf{M}}_S - \mathbf{M}^*_S\|_1.$$

Using the compatibility condition, there exists some $\psi > 0$ such that

$$\|\hat{\mathbf{M}}_S - \mathbf{M}^*_S\|_1 \leq (\sqrt{N}\psi)^{-1}\sqrt{\sum_{ij}(\hat{d}(i,j) - d^*(i,j))^2}.$$

Hence

$$\begin{aligned} 2\sum_{ij} \frac{(\hat{d}(i,j) - d^*(i,j))^2}{N} + \lambda\|\hat{\mathbf{M}} - \mathbf{M}^*\|_1 &\leq 4\lambda\frac{\sqrt{\sum_{ij}(\hat{d}(i,j) - d^*(i,j))^2}}{\sqrt{N}\psi} \\ &\leq \sum_{ij} \frac{(\hat{d}(i,j) - d^*(i,j))^2}{N} + \frac{4\lambda^2}{\psi^2}, \end{aligned}$$

which completes the proof. ∎

Chapter 3

Group selection

Yuan and Lin (2006) proposed the group lasso as the natural extension of lasso for selecting groups of variables. Their discussion is limited to non-overlapping groups, i.e. a coefficient does not involve membership in multiple groups. Zhao et al. (2009) investigated overlapping group penalization and showed that a specially constructed overlapping group structure allows hierarchical selection. In this chapter, we discuss sparsity on the number of underlying variables, i.e. assuming certain rows and columns of $\mathbf{M}$ are zero. We demonstrate that an overlapping group lasso penalty gives the necessary group selection hierarchy that recovers this type of sparsity.

This chapter is organized as follows. In Section 3.1, we highlight some of our contribution. An introduction of the group lasso and overlapping group lasso is given in Section 3.2 and 3.3. In Section 3.4, we propose our group penalty and demonstrate that the hierarchical selection property with a 2×2 matrix. In Section 3.5, we prove a sparse recovery result similar to the one given in Chapter 2. We then discuss an algorithm that allows us to perform the associated optimization problem in Section 3.6. Finally, we present some numerical simulations to compare the effectiveness of the lasso penalty in Chapter 2 and the group penalty in Section 3.7.

3.1 Our contribution

In this chapter, the sparsity assumption is that true metric $\mathbf{M}^*$ has many zero rows/columns. The motivation of this assumption is that this is equivalent to selecting the underlying variables. If a row/column is zero, then the corresponding variable is not involved in the computation of the distances.

Our contribution is to demonstrate some of the theoretical properties of the following group lasso penalty

$$\|\mathbf{M}\|_{group} = \sum_{i=1}^{p} \sqrt{\sum_{j=1}^{p} \mathbf{M}_{i,j}^2}.$$

Since $\mathbf{M}$ is symmetric, all off-diagonal entries of $\mathbf{M}$ belong to two groups. We first show that the penalty $\|.\|_{group}$ enables hierarchical selection that an off-diagonal entry is non-zero only if both of its underlying variables are selected.

We then prove a sparse recovery result similar to theorem 2.2.4 with the following form

$$N^{-1} \sum_{ij} \left[d_{\hat{\mathbf{M}}}(i,j) - d^*(i,j) \right]^2 \; + \; \lambda \|\hat{\mathbf{M}} - \mathbf{M}^*\|_{group} \; \leq \; \frac{4\lambda^2}{\psi^2}.$$

where regularization parameter λ can be chosen of the order $O(\sqrt{p \log p / n})$.

While the optimization problem remains convex, it is more difficult to solve as the non-differentiable group penalties are no longer separable. The simple group descent algorithm proposed by Yuan and Lin (2006) no longer applies. We discuss an optimization algorithm called Alternating Direction Method of Multipliers (Boyd et al., 2011). The key idea of this technique is to introduce new variables such that the penalty becomes one of a non-overlapping group lasso under the new parametrisation.

3.2 Group Lasso

The extension of lasso in group selection is first proposed by Yuan and Lin (2006) as the group lasso. Suppose the set $\{1, \ldots, p\}$ is partitioned into $G_1, \ldots, G_k$, with $G_i \cap G_j = \emptyset$. We can then write

$$\beta = (\beta_{G_1}, \ldots, \beta_{G_k}).$$

Instead of an $\ell 1$ regularization on each individual term, group lasso proposes to use the Euclidean norm of each group to promote group sparsity

$$\lambda \sum_{j]1}^{k} w_j ||\beta_{G_j}||_2$$

where $||\beta_{G_j}||_2 = \sqrt{\sum_{i \in G_j} \beta_i^2}$.

This penalty has a non-differentiable turning point at $\beta_{G_j} \equiv 0$ and hence encourages group sparsity such that the entire group of coefficients β_{G_j} will be zero. Yuan and Lin (2006) only discussed the case when the groups are disjoint. Meier et al. (2008) and Chesneau and Hebiri (2008) considered some of the theoretical properties of the non-overlapping group lasso problem. There are two extensions to cover the overlapping case we will discuss in the following section.

3.3 Overlapping group lasso

There are two different approaches to extend the group lasso to overlapping groups. Jacob et al. (2009) proposed an extension where the non-zeros are in the union of the selected groups. Zhao et al. (2009) proposed an extension where the zeros are in the union of the unselected groups, in other words, the non-zeros are in the complement of the union of the unselected groups. When the groups are disjoint, these two sparsity assumptions are equivalent: the complement of the union of the unselected groups equals to the union of the selected groups. With overlapping groups, these two sparsity assumptions differ and have different applications.

3.3.1 Graph Lasso

Jacob et al. (2009) suggested an extension of the group lasso if the non-zero entries can be represented by a union of the selected groups. For example in the following regression model

$$y = x_1\beta_1 + x_2\beta_2 + x_3\beta_3,$$

let $G_1 := \{1, 2\}, G_2 := \{2, 3\}$. The possible non-zero coefficients patterns for β are all possible union of groups: $\emptyset, G_1, G_2, G_1 \cup G_2$

The idea is to duplicate the variables such that it now becomes a non-overlapping group lasso problem. Jacob et al. (2009) proposed to fit

$$y = x_1\beta_1 + x_2\beta_{2(G_1)} + x_2\beta_{2(G_2)} + x_3\beta_3$$

with the group penalty $P(\beta) = \sqrt{\beta_1^2 + \beta_{2(G_1)}^2} + \sqrt{\beta_{2(G_2)}^2 + \beta_3^2}$. This extension has the benefit that existing consistency theories for non-overlapping group lasso immediately apply. However, one would need to specify all possible groups with this approach. In

the metric learning setting, this would be all possible sub-matrices. Since the number of sub-matrices increases exponentially with the number of p and the nested structure of our problem, we do not proceed with this approach.

3.3.2 Hierarchical Group Lasso

Zhao et al. (2009) considered overlapping groups penalty. With overlapping group penalty, the non-zeros are within the complement of the union of the unselected groups. Therefore, a set of overlapping groups imposes restrictions on the possible non-zero patterns of the fitted model which Zhao et al. (2009) called hierarchical selection. An application of hierarchical selection would be in models involving interaction terms to impose the restriction that an interaction term can be added only if the corresponding main effect coefficients are non-zero. For example in the following regression model,

$$y = x_1\beta_1 + x_2\beta_2 + x_1x_2\beta_{12}$$

we would like to enforce the restriction that β_{12} cannot be selected when β_1 or β_2 are zero. An overlapping group lasso penalty $P(\beta) = \sqrt{\beta_1^2 + \beta_{12}^2} + \sqrt{\beta_2^2 + \beta_{12}^2}$ would introduce such a hierarchical selection property. The coefficient β_{12} is located within two groups $\{\beta_1, \beta_{12}\}, \{\beta_2, \beta_{12}\}$ simultaneously. If β_{12} is non-zero, then $\sqrt{\beta_1^2 + \beta_{12}^2} > 0$ and $\sqrt{\beta_2^2 + \beta_{12}^2} > 0$ imply that the penalty $P(\beta)$ is differentiable and hence loses its sparsity inducing property. Therefore, β_{12} is non-zero only if both groups are selected. Zhao et al. (2009) further discussed how a hierarchical structure represented by a directed acyclic graph can be imposed using overlapping group Lasso and demonstrated good predictive performance via simulation experiments. The metric learning problem is similar to fitting a model with interaction terms, with each off-diagonal entry having two underlying variables as its parents. In the next section, we will illustrate how an overlapping group penalty recovers a fitted metric $\hat{\mathbf{M}}$ with zero columns/rows.

3.4 Metric Learning with Group Lasso

In the metric learning problem, we consider the group lasso with overlapping groups,

$$\min_{\mathbf{M}} L(\mathbf{M}) + ||\mathbf{M}||_{group} \tag{3.1}$$

where the penalty $||\mathbf{M}||_{group}$ for $p \times p$-dimensional matrix $\mathbf{M}$ is defined as

$$||\mathbf{M}||_{group} := \sum_{i=1}^{p} \sqrt{\sum_{j} \mathbf{M}_{i,j}^2}.$$

Since the matrix $\mathbf{M}$ is symmetric, this effectively defines a group lasso with overlapping groups. The overlapping group structure imposes a hierarchical selection pattern such that the off-diagonal entries are non-zero only if both groups corresponding to the underlying variables are selected. For example, if the top five rows are selected, only the top 5×5 block of the fitted matrix $\hat{\mathbf{M}}$ will be non-zero. We illustrate this property with a 2×2 matrix in the next section.

3.4.1 Hierarchical Selection

We illustrate the hierarchical selection property with a 2×2 matrix $\mathbf{M}$

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{pmatrix}.$$

The associated group penalized optimization problem would be

$$\min_{\mathbf{M}} L(\mathbf{M}) + \lambda \|\mathbf{M}\|_{group}$$

where the penalty $\|\mathbf{M}\|_{group} = \sqrt{m_{11}^2 + m_{12}^2} + \sqrt{m_{22}^2 + m_{12}^2}$.

Since the above optimization problem is convex, the solution is characterized by the Karush-Kuhn-Tucker conditions. As the group penalty is not differentiable when $m_{11}^2 + m_{12}^2 = 0$ or $m_{22}^2 + m_{12}^2 = 0$, we consider four different cases depending on whether $m_{11}^2 + m_{12}^2 = 0$ or $m_{22}^2 + m_{12}^2 = 0$.

Case 1 $m_{11}^2 + m_{12}^2 \neq 0$ and $m_{22}^2 + m_{12}^2 \neq 0$

The Karush-Kuhn-Tucker conditions are

$$\frac{\partial L}{\partial m_{11}} = -\lambda \frac{m_{11}}{\sqrt{m_{11}^2 + m_{12}^2}} \tag{3.2}$$

$$\frac{\partial L}{\partial m_{22}} = -\lambda \frac{m_{22}}{\sqrt{m_{22}^2 + m_{12}^2}} \tag{3.3}$$

$$\frac{\partial L}{\partial m_{12}} = -\lambda \left(\frac{m_{12}}{\sqrt{m_{11}^2 + m_{12}^2}} + \frac{m_{12}}{\sqrt{m_{22}^2 + m_{12}^2}} \right). \tag{3.4}$$

Assuming the solutions of the above equations are not $m_{11} = 0$, $m_{12} = 0$ or $m_{22} = 0$, this set of equations implies a few interesting properties. When $m_{12} \neq 0$, then m_{11} and m_{22} are non-zero. When $m_{11} \neq 0$ and $m_{22} \neq 0$, then m_{12} is non-zero.

Case 2 $m_{11}^2 + m_{12}^2 = 0$ and $m_{22}^2 + m_{12}^2 = 0$ This is the case when $\mathbf{M} = 0$.

$$\left| \frac{\partial L}{\partial m_{11}} \right| \leq \lambda \tag{3.5}$$

$$\left| \frac{\partial L}{\partial m_{22}} \right| \leq \lambda \tag{3.6}$$

$$\left| \frac{\partial L}{\partial m_{12}} \right| \leq 2\lambda \tag{3.7}$$

The group penalty has the similar thresholding properties that the coefficients are kept at zero when the gradient of the loss function $\left| \frac{\partial L}{\partial m_{11}} \right|$ is small.

Case 3 $m_{11}^2 + m_{12}^2 \neq 0$ and $m_{22}^2 + m_{12}^2 = 0$ This is the case when $m_{11} \neq 0$ and $m_{22} = m_{12} = 0$.

$$\frac{\partial L}{\partial m_{11}} = -\lambda \frac{m_{11}}{\sqrt{m_{11}^2 + m_{12}^2}} \tag{3.8}$$

$$\left| \frac{\partial L}{\partial m_{22}} \right| \leq \lambda \tag{3.9}$$

$$\left| \frac{\partial L}{\partial m_{12}} \right| \leq \lambda \tag{3.10}$$

When m_{11} is non-zero, the penalty remains non-differentiable at $m_{22} = m_{12} = 0$ and m_{12} and m_{22} can still be kept at zero by the penalty.

Case 4 $m_{11}^2 + m_{12}^2 \neq 0$ and $m_{22}^2 + m_{12}^2 = 0$ This is similar to case 3 with the m_{11} replaced by m_{22}.

Combining these four cases, we deduce that the penalty naturally encourages the following four sparsity patterns:

m_{11}	m_{12}	m_{22}
0	0	0
✓	0	0
0	0	✓
✓	✓	✓

The following pattern would only occur by chance under the group penalty, but would be encouraged by a simple entrywise lasso penalty.

m_{11}	m_{12}	m_{22}
0	✓	0
✓	✓	0
0	✓	✓
✓	0	✓

.

This argument extends naturally to a larger matrix $\mathbf{M}$. All coefficients within an unselected group are zero. The non-zero entries would therefore be within the complement of unselected groups. Therefore, all non-zero entries are located within a block matrix. For example, if the top r rows are selected, only the top $r \times r$ block of the fitted matrix $\hat{\mathbf{M}}$ would be non-zero.

3.5 Sparse Recovery

The discussion of sparse recovery is divided into three parts. We first discuss a compatibility condition that applies to our group penalty in the next section. We then state the consistency theorem in Section 3.5.2 and proofs are given in Section 3.5.3

3.5.1 Compatibility condition

Suppose the true metric $\mathbf{M}^*$ has many zero rows/columns. Let $S = \{i : \exists j \text{ such that } \mathbf{M}^*_{ij} \neq 0\}$ be the set of indices of rows that have at least one non-zero entry.

Define a decomposition of an arbitrary matrix $p \times p$ matrix $\mathbf{A} = \mathbf{A}' + \mathbf{A}''$ where each entry of $\mathbf{A}''$ is defined as

$$\mathbf{A}''_{ij} = \mathbf{A}_{ij}\mathbf{1}\{i \notin S \text{ and } j \notin S\}$$

and $\mathbf{A}'$ is defined as

$$\mathbf{A}' = \mathbf{A} - \mathbf{A}''$$

If the set S equals $1, 2, \ldots, r$, $\mathbf{A}''$ is simply the lower right $(p-r) \times (p-r)$ block of the matrix $\mathbf{A}$.

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}' \\ \hline \mathbf{A}'' \end{pmatrix}$$

We now define the compatibility condition for the group penalty case.

Assumption 3.5.1 (Compatibility Condition for group lasso) *There exists* $\psi > 0$ *such that for any symmetric matrix* $\mathbf{A}$ *with* $\|\mathbf{A}''\|_{group} \leq 3\|\mathbf{A}'\|_{group}$,

$$\|\mathbf{A}'\|_{group}^2 \leq \frac{1}{16N\psi^2} \sum_{i>j} tr\big[\mathbf{A}(X_i - X_j)(X_i - X_j)^T\big]^2.$$

Since $tr(\mathbf{AB})$ is the dot product of two $p \times p$ symmetric matrices $\mathbf{A}$ and $\mathbf{B}$, if the set of matrices $(X_i - X_j)(X_i - X_j)^T$ spans the space of $p \times p$ symmetric matrix, $\sum_{i>j} \mathrm{tr}\big[\mathbf{A}(X_i - X_j)(X_i - X_j)^T\big]^2$ must be strictly positive for any non empty symmetric matrix $\mathbf{A}$ and hence compatibility condition holds. The condition that $\|\mathbf{A}''\|_{group} \leq 3\|\mathbf{A}'\|_{group}$ allows compatibility condition to hold in some situations when the set of matrices $(X_i - X_j)(X_i - X_j)^T$ does not span the space of $p \times p$ symmetric matrix. This is similar to the compatibility condition proposed for the entrywise lasso penalty in section 2.2.2, with the group penalty instead.

3.5.2 Consistency Theorem

We again assume that the assumption of bounded error and independent error holds as in assumptions 2.2.1 and 2.2.2 on page 14.

Let $\hat{\mathbf{M}}$ minimizes the following loss function.

$$\hat{\mathbf{M}} = \mathrm{argmin}_{\mathbf{M} \succeq 0} \sum_{ij} \frac{(d(i,j) - d_{\mathbf{M}}(i,j))^2}{N} + \lambda \|\mathbf{M}\|_{group}. \tag{3.11}$$

We now state our theorem.

Theorem 3.5.2 *Let* $\hat{\mathbf{M}}$ *be the estimator in (3.11). Suppose that Assumptions 2.2.2 and 2.2.1 hold. Suppose that there exists some* $c > 0$ *such that* $|X_{i,h}| \leq c$ *for all* $h = 1, \ldots, p$, *and* $i = 1, \ldots, n$. *Let*

$$\lambda := 16c^2\delta^2 \sqrt{\frac{p(t^2 + 4\log(p) + 2\log(n))}{(n-1)/2}}.$$

Then, with probability at least $1 - 2\exp(-t^2/2)$,

$$N^{-1}\sum_{ij}\left[d_{\hat{\mathbf{M}}}(i,j) - d^*(i,j)\right]^2 \;+\; \lambda\|\hat{\mathbf{M}} - \mathbf{M}^*\|_{group} \;\leq\; \frac{4\lambda^2}{\psi^2}.$$

When compared to the consistency theorem 2.2.4 for the Lasso penalty, there is an extra factor of $\sqrt{p}$ in λ. This result gives a rate of convergence of the mean squared error and the discrepancy between $\hat{\mathbf{M}}$ and $\mathbf{M}^*$ measured by the group penalty. The mean squared error converges at the rate of $O(p\log p/n)$ while $\|\hat{\mathbf{M}} - \mathbf{M}^*\|_{group}$ converges at the rate of $O(\sqrt{p\log p/n})$.

If the compatibility condition fails, then a weaker version of the consistency still holds.

Theorem 3.5.3 *Let* $\hat{\mathbf{M}}$ *be the estimator in (3.11). Suppose that Assumptions 2.2.2 and 2.2.1 hold. Suppose that there exists some* $c > 0$ *such that* $|X_{i,h}| \leq c$ *for all* $h = 1, \ldots, p$, *and* $i = 1, \ldots, n$. *Let*

$$\lambda := 16c^2\delta^2\sqrt{p\frac{t^2 + 4\log(p) + 2\log(n)}{(n-1)/2}}.$$

Then, with probability at least $1 - 2\exp(-t^2/2)$,

$$2N^{-1}\sum_{ij}\left[d_{\hat{\mathbf{M}}}(i,j) - d^*(i,j)\right]^2 \;\leq\; 3\lambda\|\mathbf{M}^*\|_{group}.$$

If compatibility condition fails, the mean squared error still converges at a slower rate of $1/\sqrt{n}$ when compared with the rate of $1/n$ when compatibility condition holds.

3.5.3 Proofs

The decomposition $\mathbf{A} = \mathbf{A}' + \mathbf{A}''$ is an analogue for the group lasso case of the decomposition $\mathbf{A} = \mathbf{A}_S + \mathbf{A}_{S^c}$ given in 2.2.3. In the proof of theorem 2.2.4 in chapter 2, the equality $\|\mathbf{A}\|_1 = \|\mathbf{A}_S\|_1 + \|\mathbf{A}_{S^c}\|_1$ is used in a couple places. However, this equality in general does not hold under the decomposition $\mathbf{A} = \mathbf{A}' + \mathbf{A}''$ and one can only assume an inequality $\|\mathbf{A}' + \mathbf{A}''\|_{group} \leq \|\mathbf{A}'\|_{group} + \|\mathbf{A}''\|_{group}$. Therefore, a modification of the proofs is required.

It turns out that the equality $\|\mathbf{M}^* + \mathbf{A}''\|_{group} = \|\mathbf{M}^*\|_{group} + \|\mathbf{A}''\|_{group}$ is sufficient in proving the consistency result at the cost of a constant multiplicative factor. This

equality can be proved with a simple argument. If $\mathbf{M}^*$ is the top-left $r \times r$ block matrix, then $\mathbf{A}''$ is the lower-right $(p-r) \times (p-r)$ block matrix by definition.

$$\mathbf{M}^* + \mathbf{A}'' = \left(\begin{array}{c|c} \mathbf{M}^* & 0 \\ \hline 0 & \mathbf{A}'' \end{array} \right)$$

Hence, the equation $\|\mathbf{M}^* + \mathbf{A}''\|_{group} = \|\mathbf{M}^*\|_{group} + \|\mathbf{A}''\|_{group}$ holds.

The following Lemma is similar to Lemma 2.5.2.

Lemma 3.5.4 *Assume that* $\max_{1 \le h \le p, 1 \le \ell \le p} |2N^{-1} \sum_{ij} \varepsilon_{i,j}(X_{ih} - X_{jh})(X_{i\ell} - X_{j\ell})| < \lambda_0$. *Then*

$$N^{-1} \sum_{ij} (\hat{d}(i,j) - d^*(i,j))^2 + \lambda \|\hat{\mathbf{M}}\|_{group} \le \lambda_0 \sqrt{p} \|\hat{\mathbf{M}} - \mathbf{M}^*\|_{group} + \lambda \|\mathbf{M}^*\|_{group}.$$

Proof Since $\hat{\mathbf{M}}$ minimizes $L(\mathbf{M}) + \lambda \|\mathbf{M}\|_{group}$, we have $L(\hat{\mathbf{M}}) + \lambda \|\hat{\mathbf{M}}\|_{group} \le L(\mathbf{M}^*) + \lambda \|\mathbf{M}^*\|_{group}$ and thus

$$\begin{aligned}
\sum_{ij} \frac{(d(i,j) - \hat{d}(i,j))^2}{N} + \lambda \|\hat{\mathbf{M}}\|_{group} &\le \sum_{ij} \frac{(d(i,j) - d^*(i,j))^2}{N} + \lambda \|\mathbf{M}^*\|_{group} \\
-2 \sum_{ij} \frac{\hat{d}(i,j) d(i,j)}{N} + \sum_{ij} \frac{\hat{d}(i,j)^2}{N} + \lambda \|\hat{\mathbf{M}}\|_{group} &\le -2 \sum_{ij} \frac{d^*(i,j) d(i,j)}{N} + \sum_{ij} \frac{d^*(i,j)^2}{N} \\
&\quad + \lambda \|\mathbf{M}^*\|_{group} \\
\sum_{ij} \frac{(\hat{d}(i,j) - d^*(i,j))^2}{N} + \lambda \|\hat{\mathbf{M}}\|_{group} &\le \sum_{ij} 2(d(i,j) - d^*(i,j)) \frac{(\hat{d}(i,j) - d^*(i,j))}{N} \\
&\quad + \lambda \|\mathbf{M}^*\|_{group} \\
\sum_{ij} \frac{(\hat{d}(i,j) - d^*(i,j))^2}{N} + \lambda \|\hat{\mathbf{M}}\|_{group} &\le \sum_{ij} 2\varepsilon_{ij} \frac{(\hat{d}(i,j) - d^*(i,j))}{N} + \lambda \|\mathbf{M}^*\|_{group} \\
&= \frac{\sum_{ij} 2\varepsilon_{i,j} tr((\hat{\mathbf{M}} - \mathbf{M}^*)(X_i - X_j)(X_i - X_j)^T)}{N} \\
&\quad + \lambda \|\mathbf{M}^*\|_{group} \\
&= \frac{tr((\hat{\mathbf{M}} - \mathbf{M}^*) \sum_{ij} 2\varepsilon_{i,j}(X_i - X_j)(X_i - X_j)^T)}{N} \\
&\quad + \lambda \|\mathbf{M}^*\|_{group} \\
&\le \lambda_0 \|\hat{\mathbf{M}} - \mathbf{M}^*\|_1 + \lambda \|\mathbf{M}^*\|_{group} \\
&\le \lambda_0 \sqrt{p} \|\hat{\mathbf{M}} - \mathbf{M}^*\|_{group} + \lambda \|\mathbf{M}^*\|_{group},
\end{aligned}$$

where we applied the inequality $\|\mathbf{M}\|_1 \le \sqrt{p} \|\mathbf{M}\|_{group}$ in the last step. ∎

We first prove a lemma that shows that compatibility condition only needs to hold over the cone $\|\mathbf{A}''\|_{group} \leq 3\|\mathbf{A}'\|_{group}$.

Lemma 3.5.5 *Let* $\Delta = \hat{\mathbf{M}} - \mathbf{M}^*$ *and assume that* $\max_{1\leq h\leq p, 1\leq \ell\leq p} |2N^{-1}\sum_{ij} \varepsilon_{i,j}(X_{ih} - X_{jh})(X_{i\ell} - X_{j\ell})| < \lambda_0$. *If* $\lambda > 2\sqrt{p}\lambda_0$, *then* $||\Delta''||_{group} \leq 3||\Delta'||_{group}$.

Proof For a simpler notation, we shall write $\|.\|$ to stand for $\|.\|_{group}$ in this proof.

$$\begin{aligned}
||\hat{\mathbf{M}}|| &= ||\mathbf{M}^* + \Delta|| \\
&= ||\mathbf{M}^* + \Delta'' + \Delta'|| \\
&\geq ||\mathbf{M}^* + \Delta''|| - ||\Delta'|| \quad \text{by triangle inequality} \\
&= ||\mathbf{M}^*|| + ||\Delta''|| - ||\Delta'||
\end{aligned}$$

where we have used the fact that $||\mathbf{M}^* + \Delta''|| = ||\mathbf{M}^*|| + ||\Delta''||$.

By Lemma 3.5.4 and multiplying both sides by 2,

$$\begin{aligned}
2\sum_{ij} \frac{(d^*(i,j) - \hat{d}(i,j))^2}{N} + 2\lambda||\hat{\mathbf{M}}|| &\leq 2\lambda_0\sqrt{p}\|\hat{\mathbf{M}} - \mathbf{M}^*\| + 2\lambda||\mathbf{M}^*|| \\
2\sum_{ij} \frac{(d^*(i,j) - \hat{d}(i,j))^2}{N} + 2\lambda||\hat{\mathbf{M}}|| &\leq \lambda\|\hat{\mathbf{M}} - \mathbf{M}^*\| + 2\lambda\|\mathbf{M}^*\| \\
0 &\leq \lambda\|\Delta\| + 2\lambda(\|\Delta'\| - \|\Delta''\|) \\
0 &\leq \lambda\|\Delta'\| + \lambda\|\Delta''\| + 2\lambda\|\Delta'\| - 2\lambda\|\Delta''\| \\
0 &\leq 3\lambda\|\Delta'\| - \lambda\|\Delta''\|
\end{aligned}$$

where we apply the choice $\lambda > 2\sqrt{p}\lambda_0$ in the second step. ∎

A very similar argument would also give the proof of theorem 3.5.3.

Lemma 3.5.6 *Assume that* $\max_{1\leq h\leq p, 1\leq \ell\leq p} |2N^{-1}\sum_{ij} \varepsilon_{i,j}(X_{ih} - X_{jh})(X_{i\ell} - X_{j\ell})| < \lambda_0$. *If* $\lambda > 2\sqrt{p}\lambda_0$, *then* $2\sum_{ij} \frac{(d^*(i,j) - \hat{d}(i,j))^2}{N} \leq 3\lambda\|\mathbf{M}^*\|$.

Proof By Lemma 3.5.4,

$$2\sum_{ij}\frac{(d^*(i,j)-\hat{d}(i,j))^2}{N}+2\lambda||\hat{\mathbf{M}}||\leq 2\lambda_0\sqrt{p}\|\hat{\mathbf{M}}-\mathbf{M}^*\|+2\lambda||\mathbf{M}^*||$$

$$2\sum_{ij}\frac{(d^*(i,j)-\hat{d}(i,j))^2}{N}\leq \lambda\|\hat{\mathbf{M}}-\mathbf{M}^*\|+2\lambda\|\mathbf{M}^*\|-2\lambda\|\hat{\mathbf{M}}\|$$

$$2\sum_{ij}\frac{(d^*(i,j)-\hat{d}(i,j))^2}{N}\leq \lambda\|\hat{\mathbf{M}}\|+\lambda\|\mathbf{M}^*\|+2\lambda\|\mathbf{M}^*\|-2\lambda\|\hat{\mathbf{M}}\|$$

$$2\sum_{ij}\frac{(d^*(i,j)-\hat{d}(i,j))^2}{N}\leq 3\lambda\|\mathbf{M}^*\|$$

∎

The following lemma is a technical step of the final proof.

Lemma 3.5.7 *Assume that* $\max_{1\leq h\leq p,1\leq \ell\leq p}|2N^{-1}\sum_{ij}\varepsilon_{i,j}(X_{ih}-X_{jh})(X_{i\ell}-X_{j\ell})|<\lambda_0$. *If* $\lambda>2\sqrt{p}\lambda_0$, *then*

$$2\sum_{ij}\frac{(d^*(i,j)-\hat{d}(i,j))^2}{N}\leq 3\lambda||\hat{\mathbf{M}}-\mathbf{M}^*||_{group}$$

Proof By Lemma 3.5.4,

$$2\sum_{ij}\frac{(d^*(i,j)-\hat{d}(i,j))^2}{N}+2\lambda||\hat{\mathbf{M}}||_*\leq 2\sqrt{p}\lambda_0\|\hat{\mathbf{M}}-\mathbf{M}^*\|+2\lambda||\mathbf{M}^*||$$

$$2\sum_{ij}\frac{(d^*(i,j)-\hat{d}(i,j))^2}{N}+2\lambda||\hat{\mathbf{M}}||_*\leq \lambda||\hat{\mathbf{M}}-\mathbf{M}^*||+2\lambda||\mathbf{M}^*|| \text{ by the choice of } \lambda$$

$$2\sum_{ij}\frac{(d^*(i,j)-\hat{d}(i,j))^2}{N}\leq \lambda||\hat{\mathbf{M}}-\mathbf{M}^*||+2\lambda(||\mathbf{M}^*||-||\hat{\mathbf{M}}||)$$

$$2\sum_{ij}\frac{(d^*(i,j)-\hat{d}(i,j))^2}{N}\leq \lambda||\hat{\mathbf{M}}-\mathbf{M}^*||+2\lambda(||\mathbf{M}^*-\hat{\mathbf{M}}||) \text{ by triangle inequality .}$$

∎

The following Lemma is the last step of our proofs.

Lemma 3.5.8 *Assume that* $\max_{1\leq h\leq p,1\leq \ell\leq p}|2N^{-1}\sum_{ij}\varepsilon_{i,j}(X_{ih}-X_{jh})(X_{i\ell}-X_{j\ell})|<\lambda_0$. *If* $\lambda>2\sqrt{p}\lambda_0$, *then*

$$\sum_{ij}\frac{(d^*(i,j)-\hat{d}(i,j))^2}{N}+\lambda||\hat{\mathbf{M}}-\mathbf{M}^*||\leq 4\frac{\lambda^2}{\phi^2}$$

Proof By Lemma 3.5.7,

$$
\begin{aligned}
2\sum_{ij}\frac{(d^*(i,j)-\hat{d}(i,j))^2}{N}+\lambda||\hat{\mathbf{M}}-\mathbf{M}^*|| &\leq 4\lambda||\hat{\mathbf{M}}-\mathbf{M}^*|| \\
&\leq 4\lambda\|(\hat{\mathbf{M}}-\mathbf{M}^*)'+(\hat{\mathbf{M}}-\mathbf{M}^*)''\| \\
&\leq 4\lambda\left(\|(\hat{\mathbf{M}}-\mathbf{M}^*)'\|+\|(\hat{\mathbf{M}}-\mathbf{M}^*)''\|\right) \\
&\leq 16\lambda\|(\hat{\mathbf{M}}-\mathbf{M}^*)'\| \\
&\leq 4\lambda\frac{\sqrt{\sum_{ij}(d^*(i,j)-\hat{d}(i,j))^2}}{\sqrt{N}\phi} \text{ (compatibility condition)} \\
&\leq \sum_{ij}\frac{(d^*(i,j)-\hat{d}(i,j))^2}{N}+4\frac{\lambda^2}{\phi^2}.
\end{aligned}
$$

where we applied compatibility condition with the following inequality

$$\|\mathbf{A}\|_{group}^2 \leq \|\mathbf{A}'+\mathbf{A}''\|_{group}^2 \leq (\|\mathbf{A}'\|_{group}+\|\mathbf{A}''\|_{group})^2 \leq 16\|\mathbf{A}'\|_{group}^2.$$

and the inequality $3\|\Delta'\| \geq \|\Delta''\|$ in the fourth line and the inequality $4ab \leq a^2+4b^2$ in the last step. ∎

The proof of theorem 3.5.2 is completed by the choice of λ_0 in Lemma 2.5.1 and the above lemma.

3.6 Algorithm

Group lasso with overlaps is a difficult optimization problem as the non-differentiable penalties are no longer separable and we can no longer use the optimization method described in Yuan and Lin (2006). We first review the algorithm proposed in Yuan and Lin (2006) and explain why it cannot generalize to the overlapping case. We then introduce a technique called the alternating direction method of multipliers (Boyd et al., 2011) which allows us to effectively convert a problem with overlapping groups into a problem with non-overlapping groups by introducing new variables.

3.6.1 Algorithm for non-overlapping group lasso

Suppose the set $\{1,\ldots,p\}$ is partitioned into $G_1,\ldots,G_k$, with $G_i \cap G_j = \emptyset$ and the set of coefficients are $\beta = (\beta_{G_1},\ldots,\beta_{G_k})$. For an optimization with a smooth convex

loss function L and group lasso penalty

$$\min_{\beta} L(\beta) + \lambda \sum_{j=1}^{k} \|\beta_{G_j}\|_2,$$

the solution is characterized by the Karush–Kuhn–Tucker conditions:

$$\begin{aligned} \frac{\partial L}{\partial \beta_{G_j}} + \lambda \frac{\beta_{G_j}}{\|\beta_{G_j}\|_2} = 0 \quad \forall \beta_{G_j} \neq 0, \\ \left\| \frac{\partial L}{\partial \beta_{G_j}} \right\|_2 \leq \lambda \quad \forall \beta_{G_j} = 0. \end{aligned}$$

Solution to the above conditions can be combined into a single equation as

$$\beta_{G_j} = \left(1 - \frac{\lambda}{\left\| \frac{\partial L}{\partial \beta_{G_j}} \right\|_2} \right) \frac{\partial L}{\partial \beta_{G_j}}. \tag{3.12}$$

Yuan and Lin (2006) suggested that the solution can be computed by iteratively applying (3.12) until convergence. Similar alogrithm was also proposed by Fu (1998) to solve lasso regularized problem.

However, if there are overlapping groups, for example the 2×2 case described in Section 3.4, the Karush–Kuhn–Tucker conditions no longer have the simple form of equation (3.12). Therefore, an alternative algorithm is required to solve the overlapping group lasso problem.

3.6.2 Alternating direction method of multipliers

In the last couple of years, a technique called the alternating direction method of multipliers (Boyd et al., 2011) has been used to solve many of the $\ell 1$ optimization problems. In order to solve a typical $\ell 1$-regularization problem

$$\min_{\beta} L(\beta) + \Omega(\beta) \tag{3.13}$$

where L is a differentiable loss function and Ω consists of non-differentiable penalties, the key idea is to introduce extra variables γ, by linear equations $\gamma = \mathbf{A}\beta$,

$$\min_{\beta,\gamma} L(\beta) + \Omega(\gamma) \text{ and } \mathbf{A}\beta = \gamma$$

such that the optimization is simpler under the new parametrisation γ. For example, an overlapping group penalty under β becomes a non-overlapping one under the

new parametrisation γ. This decoupling allows us to easily handle different types of non-differentiable penalties and positive semi-definite constraints.

In order to enforce the equality constraints $\mathbf{A}\beta = \gamma$, we consider the augmented Lagrangian

$$L(\beta, \gamma, \nu) = L(\beta) + \nu^T(\gamma - \mathbf{A}\beta) + \frac{1}{2\mu}||\mathbf{A}\beta - \gamma||^2 + \Omega(\gamma).$$

Solving the optimization problem (3.13) is transformed into a problem of finding a saddle point of the above augmented Lagrangian.

We now introduce the algorithm in more generalized notation. Consider the following optimization problem

$$\min f(\beta) + g(\gamma) \text{ subject to } \mathbf{A}\beta + \mathbf{B}\gamma = c. \tag{3.14}$$

The problem of solving equation 3.14 is equivalent to finding a saddle point of the Lagrangian. The Lagrangian of the problem is

$$L_0(\beta, \gamma, \nu) = f(\beta) + g(\gamma) + \nu^T(\mathbf{A}\beta + \mathbf{B}\gamma - c).$$

where ν is the dual variables.

The Lagrangian is augmented by $\rho/2||\mathbf{A}\beta + \mathbf{B}\gamma - c||_2^2$.

$$L_p(\beta, \gamma, \nu) = f(\beta) + g(\gamma) + \nu^T(\mathbf{A}\beta + \mathbf{B}\gamma - c) + \frac{\rho}{2}||\mathbf{A}\beta + \mathbf{B}\gamma - c||_2^2.$$

A saddle point of the original Lagrangian is a saddle point of the augmented Lagrangian. The extra squared error term makes the Lagrangian slightly more stable from the optimization point of view.

The alternating direction method of multipliers is then performed by iteratively solving the following equations.

$$\begin{aligned}
\beta^{k+1} &:= \text{argmin}_\beta L_p(\beta, \gamma^k, \nu^k) \\
\gamma^{k+1} &:= \text{argmin}_\gamma L_p(\beta^{k+1}, \gamma, \nu^k) \\
\nu^{k+1} &:= \nu^k + \rho(\mathbf{A}\beta + \mathbf{B}\gamma - c)
\end{aligned}$$

The last step can be seen as a dual ascent step with a stepsize of ρ. The choice of ρ is not critical and the algorithm converges under quite general conditions.

3.6.3 An alternative representation

It is possible to obtain an alternative representation of the iterations by reparametrisation.

Let $r = \mathbf{A}\beta + \mathbf{B}\gamma - c$.

$$\begin{aligned}\nu^T(\mathbf{A}\beta + \mathbf{B}\gamma - c) + \frac{\rho}{2}||\mathbf{A}\beta + \mathbf{B}\gamma - c||_2^2 &= \nu^T r + \frac{\rho}{2}||r||_2^2 \\ &= \frac{\rho}{2}||r + u||_2^2 - \frac{\rho}{2}||u||_2^2\end{aligned}$$

where $u = \nu/\rho$.

The iterations can be rewritten as

$$\begin{aligned}\beta^{k+1} &:= \text{argmin}_\beta f(\beta) + \frac{\rho}{2}||\mathbf{A}\beta + \mathbf{B}\gamma^k - c + u^k||_2^2 \\ \gamma^{k+1} &:= \text{argmin}_\gamma g(\gamma) + \frac{\rho}{2}||\mathbf{A}\beta^{k+1} + \mathbf{B}\gamma - c + u^k||_2^2 \\ u^{k+1} &:= u^k + \mathbf{A}\beta^{k+1} + \mathbf{B}\gamma^{k+1} - c\end{aligned}$$

Therefore, the β and γ updates becomes a problem of minimizing f(or g) with a squared error loss term. For example, if f is a smooth loss function, g is the lasso penalty $\|\gamma\|_1$ and $\mathbf{B}$ is chosen to be an identity matrix $\mathbf{B} = \mathbf{I}$, the β update is now a minimization of the sum of two smooth loss functions and the γ update is a lasso problem on an orthogonal set of basis which can be computed easily. More details of solving the lasso problem will be given in Section 3.6.7. It is also interesting to note that the quantity u^k has an interpretation as the sum of primal residuals

$$u^k = u^0 + \sum_{j=0}^{k} r^j.$$

3.6.4 Stopping criterion

We terminate the algorithm and return the result when the iterate is sufficiently close to the optimal solution. At the optimal point, both primal and dual feasibility conditions must be satisfied. The condition for primal feasibility is

$$\mathbf{A}\beta^* + \mathbf{B}\gamma^* - c = 0, \tag{3.15}$$

while the conditions for dual feasibility are

$$0 \in \partial f(\beta^*) + \mathbf{A}^T \nu^*, \tag{3.16}$$

$$0 \in \partial g(\gamma^*) + \mathbf{B}^T \nu^*. \tag{3.17}$$

We now show that the iterate satisfies (3.17) after every iteration.

$$\begin{aligned}
\partial g(\gamma^{k+1}) + \mathbf{B}^T \nu^{k+1} &= \partial g(\gamma^{k+1}) + \mathbf{B}^T(\nu^k + \rho r^{k+1}) \\
&= \partial g(\gamma^{k+1}) + \mathbf{B}^T \nu^k + \rho \mathbf{B}^T r^{k+1} \\
&= \left. \frac{\partial L_p(\beta^{k+1}, \gamma, \nu^k)}{\partial \gamma} \right|_{\gamma=\gamma^{k+1}} \\
&= 0
\end{aligned}$$

The last step is true because γ^{k+1} minimizes $L_p(\beta^{k+1}, \gamma, \nu^k)$.

Hence, we only need to check the feasibility gap of equation 3.15 and 3.16. We first show that

$$s^{k+1} = \rho \mathbf{A}^T \mathbf{B}(\gamma^{k+1} - \gamma^k) \tag{3.18}$$

can be seen as the dual residuals of equation 3.16 by the following calculation

$$\begin{aligned}
\left. \frac{\partial L_p(\beta, \gamma^k, \nu^k)}{\partial \beta} \right|_{\beta=\beta^{k+1}} &= \partial f(\beta^{k+1}) + \mathbf{A}^T \nu^k + \rho \mathbf{A}^T(\mathbf{A}\beta^{k+1} + \mathbf{B}\gamma^k - c) \\
&= \partial f(\beta^{k+1}) + \mathbf{A}^T(\nu^k + \rho r^{k+1} + \rho \mathbf{A}^T \mathbf{B}(\gamma^k - \gamma^{k+1}) \\
&= \partial f(\beta^{k+1}) + \mathbf{A}^T \nu^{k+1} + \rho \mathbf{A}^T \mathbf{B}(\gamma^k - \gamma^{k+1})
\end{aligned}$$

$0 \in \left. \frac{\partial L_p(\beta, \gamma^k, \nu^k)}{\partial \beta} \right|_{\beta=\beta^{k+1}}$ as β^{k+1} minimizes $L_p(\beta, \gamma^k, \nu^k)$. The primal residuals are

$$r^k = \mathbf{A}\beta + \mathbf{B}\gamma - c.$$

s^k and r^k gives an indication whether the algorithm converges. We terminate the algorithm and return the result when the magnitudes of both s^k and r^k are small.

3.6.5 Convergence property

This method converges under quite general circumstances. Boyd et al. (2011) proved the following convergence theorem:

Theorem 3.6.1 *If the following conditions are met*

1. *f, g are closed, proper and convex,*

2. *L_0 has a saddle point.*

then the following holds

1. $r^k \to 0$,

2. $f(\beta^k) + g(\gamma^k) \to f(\beta^*) + g(\gamma^*)$,

3. $\nu^k \to \nu^*$.

By assuming further that $f + g$ has unique optimal point, this shows that the iterates β^k and γ^k converge to the optimal point β^* and γ^*. ∎

A huge family of loss functions, constraints and $\ell 1$ penalties satisfy above conditions, In particular, the positive semi-definite constraint, lasso penalty, group lasso penalty and the trace norm penalty satisfy them.

3.6.6 Application

The simplicity of this algorithm has led to many application in numerous optimization problems, in particular those with non-differentiable $\ell 1$-type regularization. The initial interest focused on fused lasso and total variation regularization (Goldstein and Osher, 2009). The fused lasso penalty (Tibshirani et al., 2005) in 1-dimension has the form $\Omega(\beta) = \sum_{i=1}^{p-1} |\beta_{i+1} - \beta_i|$. The fused lasso problem has a simple pathwise algorithm in 1-dimension (Friedman et al., 2007). However, the solution path of fused lasso in higher dimensions is complicated (Hoefling, 2010) and hence remain a difficult optimization problem. As this technique decouples the regularization and the regression problem, this has become one of the simplest algorithms in solving higher dimensional fused lasso problems. Boyd et al. (2011) gave a more in-depth discussion about other application of this algorithms.

We found that this decoupling is also useful in the metric learning setting. This algorithm allows us to handle positive semi-definiteness constraints, lasso and overlapping group lasso penalties in a simple way.

3.6.7 Lasso penalty

We begin with an example of how to solve a lasso problem with this algorithm, where the optimization problem is of the form

$$\min_{\beta} L(\beta) + \lambda||\beta||_1$$

New variables γ are introduced so that the problem becomes

$$\min_{\beta,\gamma} L(\beta) + \lambda||\gamma||_1 \text{ such that } \beta - \gamma = 0$$

This is equivalent to solving for a saddle point of the following Lagrangian

$$L_p(\beta, \lambda, \nu) = L(\beta) + \lambda\|\gamma\|_1 + \nu^T(\beta - \gamma).$$

We then apply the transformation given in section 3.6.3 by substituting $u = \mu/\rho$. The algorithm gives the following iterates

$$\begin{aligned}
\beta^{k+1} &:= \text{argmin}_\beta L(\beta) + \frac{\rho}{2}||\beta - \gamma^k + u^k||_2^2 \\
\gamma^{k+1} &:= \text{argmin}_\gamma \lambda||\gamma||_1 + \frac{\rho}{2}\|\beta^{k+1} - \gamma^k + u^k\|_2^2 \\
u^{k+1} &:= u^k + \beta^{k+1} - \gamma^{k+1}
\end{aligned}$$

The part involving the $\ell 1$ penalty is now in the γ update step. This is a lasso problem with orthogonal basis. The γ update step is thus simply a soft thresholding step and can be written as:

$$\gamma^{k+1} = S_{\lambda/\rho}(\beta^{k+1} + u^k)$$

where S is the soft thresholding operator

$$S_\lambda(x) = \begin{cases} x - \lambda & x > \lambda \\ 0 & |x| \leq \lambda \\ x + \lambda & x < -\lambda \end{cases}.$$

3.6.8 Group Lasso

We first consider a group lasso problem where the groups do not overlap and the extension to the overlapping case will be discussed later.

Suppose the set $\{1, \ldots, p\}$ is partitioned into $G_1, \ldots, G_k$, with $G_i \cap G_j = \emptyset$.

$$\min_{\beta} L(\beta) + \lambda \sum_{i=1}^{N} ||\beta_{G_i}||_2.$$

The idea is again to introduce new variables γ and solve the following equivalent form of the problem:

$$\min_{\beta,\gamma} L(\beta) + \lambda \sum_{i=1}^{N} ||\gamma_{G_i}||_2 \text{ such that } \beta_{G_i} - \gamma_{G_i} = 0.$$

The algorithm gives the following iterates

$$\begin{aligned}
\beta^{k+1} &:= \text{argmin}_{\beta} L(\beta) + \frac{\rho}{2}||\beta - \gamma^k + u^k||_2^2 \\
\gamma^{k+1} &:= \text{argmin}_{\gamma} \lambda \sum_{i=1}^{N} ||\gamma_{G_i}||_2 + \frac{\rho}{2}\|\beta^{k+1} - \gamma^k + u^k\|_2^2 \\
u^{k+1} &:= u^k + \beta^{k+1} - \gamma^{k+1}
\end{aligned}$$

The β and u updates are similar to the lasso penalty case. The γ update can be written as

$$\gamma_{G_i}^{k+1} := S_{\lambda/\rho}^{G}(\beta_{G_i}^{k+1} + u_{G_i}^{k}).$$

where S is now the group soft threshold operator

$$S_{\lambda}^{G}(a) = \left(1 - \frac{\lambda}{||a||_2}\right)_{+} a$$

with $S_{\lambda}^{G}(0) = 0$.

We now consider the overlapping group lasso problem. The group lasso problem with overlapping group becomes a non-overlapping group lasso problem in γ after introducing an appropriate choice of extra variables γ. The condition $\beta_{G_i} - \gamma_{G_i} = 0$ can be rewritten as $\mathbf{B}\beta - \gamma = 0$ where $\mathbf{B}$ is a matrix with $\sum |G_i|$ rows and p columns. This provides an alternative to the group descent method described in Yuan and Lin (2006).

3.6.9 Positive semi-definiteness

Solving the metric learning optimization problem requires maintaining the positive semi-definiteness of $\mathbf{M}$. This involves solving the optimization problem of the form of

$$\min_{\mathbf{M}} L(\mathbf{M}) \text{ subject to } \mathbf{M} \succeq 0$$

where $\mathbf{M} \succeq 0$ means that $\mathbf{M}$ is positive semi-definite.

We again introduce new matrix variable $\mathbf{A}$ and rewrite the above optimization as

$$\min_{\mathbf{M}} L(\mathbf{M}) + g(\mathbf{A}) \text{ such that } \mathbf{M} - \mathbf{A} = 0 \text{ and } \mathbf{A} \succeq 0.$$

The algorithm gives the following updates

$$\begin{aligned}
\mathbf{M}^{k+1} &:= \text{argmin}_{\mathbf{M}} L(\mathbf{M}) + \frac{\rho}{2}||\mathbf{M} - \mathbf{A}^k + u^k||_F^2, \\
\mathbf{A}^{k+1} &:= \text{argmin}||\mathbf{M}^{k+1} + u^k - \mathbf{A}||_F^2 \text{ subject to } \mathbf{A} \succeq 0, \\
u^{k+1} &:= u^k + \mathbf{M}^{k+1} - \mathbf{A}^{k+1}.
\end{aligned}$$

The $\mathbf{M}$ update now becomes a simple unconstrained least squares problem. The $\mathbf{A}$ update involves a projection back to the positive semi-definite cone. This projection can be computed by an eigendecomposition.

A spectral decomposition of $\mathbf{M}^{k+1}+u^k$ would give $\mathbf{M}^{k+1}+u^k = V\text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)V^T$ where V is the matrix of eigenvectors and λ the corresponding eigenvalues.

The projection onto the positive semi-definite cone would be

$$\mathbf{A}^{k+1} := V\text{diag}(\max(0, \lambda_1), \max(0, \lambda_2), \ldots, \max(0, \lambda_p))V^T.$$

A simple proof of this result can be seen in Higham (1988).

3.6.10 Combining all parts

We have demonstrated how we could handle both lasso and group lasso penalties and maintain positive semi-definiteness of the solution with the alternating directions method of multipliers. By constructing a suitable γ, we can easily combine different kinds of penalties while each update remains simple to compute. The main limitation of this algorithm is that it can only handle a relatively small $p < 100$.

By introducing new matrix variables $\mathbf{L}$, $\mathbf{G}$ and $\mathbf{P}$, the optimization problem

$$\text{argmin}_{\mathbf{M}} L(\mathbf{M}) + \lambda\|\mathbf{M}\|_1 + \lambda_{group}\|\mathbf{M}\|_{group} \text{ such that } \mathbf{M} \text{ is positive semi-definite}$$

can be transformed into

$$\text{argmin}_{\mathbf{M}} L(\mathbf{M}) + \lambda\|\mathbf{L}\|_1 + \lambda_{group}\|\mathbf{G}\|_{group} \text{ such that } \mathbf{P} \text{ is positive semi-definite}$$

with $\mathbf{M} = \mathbf{L}$, $\mathbf{M} = \mathbf{G}$ and $\mathbf{M} = \mathbf{P}$.

The outline of the algorithm is given in the following table.

Algorithm	
Initialization:	Let $\mathbf{M}^0$, $\mathbf{L}^0$, $\mathbf{G}^0$, $\mathbf{P}^0$, $\mathbf{U}_\mathbf{L}^0$, $\mathbf{U}_\mathbf{G}^0$ and $\mathbf{U}_\mathbf{P}^0$ be $p \times p$ zero matrices
Loop:	$\mathbf{M}^{k+1} := \text{argmin}_\mathbf{M} L(\mathbf{M}) + \frac{\rho}{2}\|\mathbf{M} - \mathbf{L}^k + \mathbf{U}_\mathbf{L}^k\|_F^2 + \frac{\rho}{2}\|\mathbf{M} - \mathbf{G}^k + \mathbf{U}_\mathbf{G}^k\|_F^2$ $+\frac{\rho}{2}\|\mathbf{M} - \mathbf{P}^k + \mathbf{U}_\mathbf{P}^k\|_F^2$,
	$\mathbf{L}^{k+1} := \text{argmin}_\mathbf{L} \frac{\rho}{2}\|\mathbf{M}^{k+1} + \mathbf{U}_\mathbf{L}^k - \mathbf{L}\|_F^2 + \lambda\|\mathbf{L}\|_1$,
	$\mathbf{G}^{k+1} := \text{argmin}_\mathbf{G} \frac{\rho}{2}\|\mathbf{M}^{k+1} + \mathbf{U}_\mathbf{G}^k - \mathbf{G}\|_F^2 + \lambda\|\mathbf{G}\|_{group}$,
	$\mathbf{P}^{k+1} := \text{argmin}_\mathbf{P}\|\mathbf{M}^{k+1} + \mathbf{U}_\mathbf{P}^k - \mathbf{P}\|_F^2$ such that $\mathbf{P} \succeq 0$,
	$\mathbf{U}_\mathbf{L}^{k+1} := \mathbf{U}_\mathbf{L}^{k+1} + \mathbf{M}^{k+1} - \mathbf{L}^{k+1}$,
	$\mathbf{U}_\mathbf{G}^{k+1} := \mathbf{U}_\mathbf{G}^{k+1} + \mathbf{M}^{k+1} - \mathbf{G}^{k+1}$,
	$\mathbf{U}_\mathbf{P}^{k+1} := \mathbf{U}_\mathbf{P}^{k+1} + \mathbf{M}^{k+1} - \mathbf{P}^{k+1}$.
Termination:	Terminate when the primal and dual residuals are small.

The $\mathbf{L}^{k+1}$ update step involves an entrywise soft-thresholding step described in Section 3.6.7. The $\mathbf{G}^{k+1}$ update step involves a group soft-thresholding step described in Section 3.6.8. The $\mathbf{P}^{k+1}$ update step consists of a projection computed via eigendecomposition of $\mathbf{M}^{k+1} + \mathbf{U}_\mathbf{P}^k$ described in Section 3.6.9. The primal residuals consists of three parts $\mathbf{M}^k - \mathbf{L}^k$, $\mathbf{M}^k - \mathbf{G}^k$ and $\mathbf{M}^k - \mathbf{P}^k$. The dual residuals can be computed using the formula 3.18 detailed in Section 3.6.4. The algorithm terminates when the primal and dual residuals are small.

3.7 Numerical experiments

In this section, we compare the performance of lasso and the overlapping group lasso in metric learning. Since group sparsity can be seen as a special case of entrywise sparsity, it would be interesting to compare their performance with simulations.

The setup for the simulation is detailed below.

1. Select parameters covariance matrix Σ, underlying metric $\mathbf{M}^*$, dimensionality p and noise level σ.

2. Generate independent p-dimensional $X_i \sim N(0, \Sigma), i = 1, \ldots, 300$.

3. Simulate 500 independent distances with the distribution

$$d(h) = (X_{i(h)} - X_{j(h)})^T \mathbf{M}^* (X_{i(h)} - X_{j(h)}) + \varepsilon_h, h = 1, \ldots, 500.$$

   where $\varepsilon_m \sim N(0, \sigma^2)$, $i(h), j(h)$ are drawn uniformly from any integer between 1 and 300.

4. A further 500 independent distances are simulated as a validation set to decide on the appropriate level of regularization. Finally, another 500 independent distances are generated to serve as a test set to compute an estimation of the mean squared error.

5. The model is fitted with the training data and the regularization parameter selected by the value that minimizes the squared error loss of the validation data.

6. The mean squared error of the test set is recorded.

7. Step 2-6 are repeated 50 times.

3.7.1 Simulation Detail

It remains to specify the parameters the covariance matrix Σ, the metric $\mathbf{M}^*$, the dimension p and the noise level σ. In the section, Σ is chosen to be the identity matrix and σ equals 5.

We would like to specify $\mathbf{M}^*$ to be a group sparse matrix with k non-zero rows where k is a parameter chosen to be $1 \leq k \leq p$. A group sparse metric $\mathbf{M}^*$ is generated with the following procedure. Let $\mathbf{A}$ be a $p \times p$ matrix with entries distributed as

$$\mathbf{A}_{i,j} \sim N(0, 1) \text{ if } i \leq k \text{ and } j \leq k$$
$$\mathbf{A}_{i,j} = 0 \text{ otherwise.}$$

Let

$$\mathbf{M}^* = \mathbf{A}^T \mathbf{A},$$

therefore, $\mathbf{M}^*$ has the top-left $k \times k$ block non-zero, and the rest of $\mathbf{M}^*$ is zero.

We perform simulations with different values of the dimension p and number of non-zero rows k. The following table reports the number of times the each type of penalty outperforms the other under mean squared error out of 50 simulations and the p-value of a sign test under the hypothesis that lasso perform better is given as a reference. We also report the average mean squared error differences calculated by the following formula: mean squared error of lasso - mean squared error using group penalty. Therefore, a positive value implies group penalty perform better on average. The quantities in brackets are the standard deviation of the differences.

p	k	σ	Lasso	Group Lasso	p-value	Average MSE differences
3	3	5	**34**	16	0.00767	-0.0145(0.0551)
5	3	5	**33**	17	0.0164	-0.0200(0.0618)
10	3	5	**45**	5	2.10×10^{-9}	-0.1442(0.1366)
10	8	5	**37**	13	0.000468	-0.0754(0.125)
20	3	5	**49**	1	4.53×10^{-14}	-0.393(0.204)
20	8	5	**50**	0	8.88×10^{-16}	-0.822(0.337)

There is evidence that lasso outperforms group lasso on average in every case, despite the stronger assumption that the group lasso places. Our simulation agrees with the findings in Huang and Zhang (2010) that group lasso is less effective when there is zero within group. They also hinted in their discussion that overlapping groups might have poor performance. More simulation results will be presented later in Section 4.7 to compare with the trace norm penalty in the next chapter.

3.8 Discussion

We have shown that the overlapping group lasso penalty allows sparse recovery of a distance metric $\mathbf{M}$ with zero rows/columns. However, empirical simulations showed that the overlapping group lasso is worse than the lasso penalty in terms of predictive performance. Huang and Zhang (2010) hinted that overlapping lasso does not perform very well and our simulations agreed with their claim. One explanation is that only a small proportion of coefficients in each group is actually non-zero. Some further works include investigating whether there is an alternative procedure that outperform the lasso. Forward-backward algorithms (Zhang, 2011) could possibly provide such a guarantee.

Chapter 4

Learning a low rank metric via trace norm regularization

In this chapter, we consider sparsity in the rank of $\mathbf{M}$. The rank of a symmetric matrix $\mathbf{M}$ equals to the number of non-zero eigenvalues. Since penalizing with the number of rank involves a non-convex optimization problem, we consider the trace norm, which penalizes the sum of singular values of $\mathbf{M}$, as a convex relaxation of the rank penalty. A version of sparse recovery property similar to theorem 2.2.4 proved in Chapter 2 will be proved.

This chapter is organized as follows. We introduce trace norm penalty in Section 4.2 and make the connection with the group penalty in Section 4.3. We state and prove a sparse recovery result in Section 4.4. In Section 4.5, we apply the Alternating Direction Method of Multiplier method to compute the corresponding optimization and present results from numerical simulation in Section 4.7. In Section 4.8, we discuss how metric learning problem and the matrix completion problem (Candès and Recht, 2009) can be seen as special cases of a more general regression problem. In Section 4.9, we discuss the Johnson Lindenstrauss Lemma (Johnson and Lindenstrauss, 1984) which proves there exists a distance preserving low rank projection of the data.

4.1 Matrix norm notation

Different types of matrix norm appear in this chapter. We note that some of the notations we use might stand for different matrix norms in other publications, for

example Golub and Van Loan (2012). To avoid confusion, we now define the matrix norm notation in this section. The three types of matrix norm penalties are:

$\|M\|_1 = \sum_{i=1}^{p} \sum_{j=1}^{p} |\mathbf{M}_{i,j}\|$ is the entrywise lasso penalty introduced in Chapter 2.

$\|\mathbf{M}\|_{group} = \sum_{i=1}^{p} \sqrt{\sum_{j=1}^{p} \mathbf{M}_{i,j}^2}$ is the group penalty introduced in Chapter 3.

$\|\mathbf{M}\|_* =$ **sum of singular values of M** is the trace norm penalty we are going to introduce in this chapter.

There are also some other norm that appears in the proofs and other discussions.

$\|\mathbf{M}\|_2 = \sqrt{\sum_{i=1}^{p} \sum_{j=1}^{p} \mathbf{M}_{i,j}^2}$ is the Frobenius norm.

$\|\mathbf{M}\|_\infty = \max_{i,j} |\mathbf{M}_{i,j}|$ is the entrywise infinity norm.

$\|\mathbf{M}\|_{op} =$ **the largest singular value M** is the operator norm.

With the notation now detailed, we are now ready to discuss the motivation of a trace norm penalty.

4.2 Trace norm penalty

The motivation of applying a trace norm penalty is to deal with the sparsity assumption that $\mathbf{M}^*$ is group sparse after an unknown orthogonal transformation. In fact, a low rank assumption is equivalent to the assumption that $\mathbf{M}^*$ is group sparse after an unknown orthogonal transformation. This relationship can be seen via an eigendecomposition of the matrix $\mathbf{M}$

$$\mathbf{M} = \mathbf{V}\mathbf{D}\mathbf{V}^T$$

where $\mathbf{D}$ is a diagonal matrix of eigenvalues of $\mathbf{M}$ and the columns of $\mathbf{V}$ are the corresponding eigenvectors. A matrix of rank r has r non-zero eigenvalues of $\mathbf{M}$. Such matrix can represented by non-zero eigenvalues and the corresponding eigenvectors

$$\mathbf{M} = \sum_{i=1}^{r} D_i V_i V_i^T.$$

Hence, assuming that the true metric $\mathbf{M}^*$ having a low rank restricts the number of degrees of freedom. This suggests an optimization of the following form

$$\text{argmin}_{\mathbf{M}} L(\mathbf{M}) \text{ such that } \text{rank}(\mathbf{M}) \leq r.$$

However, penalizing the number of rank is a difficult optimization problem. We consider the trace norm as a convex alternative to the rank constraint. The trace norm equals to the sum of singular values of $\mathbf{M}$.

$$\text{argmin}_{\mathbf{M}} L(\mathbf{M}) + \lambda ||\mathbf{M}||_*$$

where $||\mathbf{M}||_* = tr(\mathbf{M})$ denote the trace norm of $\mathbf{M}$, which is the sum of the singular values of $\mathbf{M}$. We will show how the trace norm penalty can be derived the group penalty in the next section. This penalty has also been used in the matrix completion problem (Candès and Plan, 2010) to introduce a low rank constraint and some of its theoretical properties of this penalty in low rank recovery have been studied by Bach (2008) and Negahban and Wainwright (2011).

4.3 Connection with group penalty

Intuitively, the assumption that $\mathbf{M}$ has low rank is equivalent to assuming that the matrix $\mathbf{M}$ has zero rows/columns after a suitable orthogonal transformation. This motivates a penalized regression of the following form

$$\min_{\mathbf{M}\succeq 0, \mathbf{U}\in O} L(\mathbf{M}) + \lambda ||\mathbf{U}^T\mathbf{M}\mathbf{U}||_{group} \tag{4.1}$$

where O is the set of all orthogonal matrices and $\mathbf{U}$ is a $p \times p$ orthogonal matrix. In this section, we present a result from Huang et al. (2009) that proved the above formulation is actually equivalent to regularization with trace norm

$$\min_{\mathbf{M}\succeq 0} L(\mathbf{M}) + \lambda \text{ tr}(\mathbf{M}).$$

The idea of the proof is to show that $tr(\mathbf{M})$ is a lower bound of $||\mathbf{U}^T\mathbf{M}\mathbf{U}||_{group}$ and this lower bound is attained when the columns of $\mathbf{U}$ are the eigenvectors of $\mathbf{M}$. In short, our aim is to prove that for any positive semi-definite $\mathbf{M}$

$$\min_{\mathbf{U}\in O} ||\mathbf{U}^T\mathbf{M}\mathbf{U}||_{group} = tr(\mathbf{M}).$$

The original proof given in Huang et al. (2009) does not cover the sparse case. An extension to their proof using a limiting argument will be presented at the end.

Proof The equation $(\sum |a_i|)^2 = \sum_i a_i^2/\gamma_i$, where $\gamma_i = |a_i|/\|a\|_1$, is used in the first line. a_i is chosen to be the $\ell 2$ norm of the i-th row of $\mathbf{U}^T\mathbf{M}\mathbf{U}$, i.e. $a_i = \|(\mathbf{U}^T\mathbf{M}\mathbf{U})_{i.}\|_2$. Let $diag(\gamma^{-1})$ be a $p \times p$ diagonal matrix with diagonal entries $\gamma_1^{-1}, \gamma_2^{-1}, \ldots, \gamma_p^{-1}$.

$$\begin{aligned}
\|\mathbf{U}^T\mathbf{M}\mathbf{U}\|_{group}^2 &= \sum_i \frac{\|(\mathbf{U}^T\mathbf{M}\mathbf{U})_{i.}\|_2^2}{\gamma_i} \\
&= tr((\mathbf{U}^T\mathbf{M}\mathbf{U})^T(\mathbf{U}^T\mathbf{M}\mathbf{U})diag(\gamma^{-1})) \\
&= tr(\mathbf{M}^T\mathbf{M}\mathbf{U}diag(\gamma^{-1})\mathbf{U}^T) \\
&= tr(\mathbf{M}^T\mathbf{M}\mathbf{V}^{-1}) \\
&= tr(\mathbf{M}^2\mathbf{V}^{-1})
\end{aligned}$$

where $\mathbf{V} = \mathbf{U}diag(\gamma)\mathbf{U}^T$ has the following properties: $tr(\mathbf{V}) = \sum \gamma_i = 1$ and $\mathbf{V}$ is positive semi-definite.

Let

$$\mathbf{A} = \mathbf{M}\mathbf{V}^{-1/2},$$

$$\mathbf{B} = \mathbf{V}^{1/2}$$

and apply Cauchy Schwarz inequality

$$|\langle \mathbf{A}, \mathbf{B} \rangle|^2 \leq \langle \mathbf{A}, \mathbf{A} \rangle \langle \mathbf{B}, \mathbf{B} \rangle$$

to obtain

$$|tr(\mathbf{M})|^2 \leq tr(\mathbf{V})tr(\mathbf{M}^2\mathbf{V}^{-1}).$$

Hence, $|tr(\mathbf{M})|$ is a lower bound of the penalization term $\|\mathbf{U}^T\mathbf{M}\mathbf{U}\|_{group}$ and the minimum is attained when the Cauchy Schwarz inequality becomes equality for $\mathbf{V} = \mathbf{M}/tr(\mathbf{M})$. This completes the proof given in Huang et al. (2009).

However, all the γ_i's are assumed to be strictly positive in the above proof. If $\mathbf{M}$ is of low rank, then some of the rows of $\mathbf{U}^T\mathbf{M}\mathbf{U}$ would be zero. This implies that some of the γ_i's would be exactly zero and hence $\mathbf{V}^{-1}$ does not exist.

This can be fixed by a limit argument. Suppose $\epsilon > 0$, let

$$\gamma_i'(\epsilon) = \begin{cases} \gamma_i & \text{if } \gamma_i \neq 0 \\ \varepsilon & \text{otherwise} \end{cases}$$

and let $\gamma'(\varepsilon) = (\gamma_1'(\epsilon), \ldots, \gamma_p'(\varepsilon))$. Define $\mathbf{V}' = \mathbf{U}diag(\gamma')\mathbf{U}^T$. The perturbed matrix $\mathbf{V}'$ is now invertible and hence Cauchy Schwarz would again imply

$$|tr(\mathbf{M})|^2 \leq tr(\mathbf{V}')tr(\mathbf{M}^2\mathbf{V}'^{-1}).$$

It remains to take the limit of ε to zero from above to obtain the required result. Since $1 \leq tr(\mathbf{V}') \leq 1 + p\varepsilon$, $tr(\mathbf{V}') \to 1$ as ε approaches 0. The second term $tr(\mathbf{M}^2\mathbf{V}'^{-1})$ does not depend on ε.

$$\begin{aligned} tr(\mathbf{M}^2\mathbf{V}'^{-1}) &= tr((\mathbf{U}^T\mathbf{M}\mathbf{U})^T(\mathbf{U}^T\mathbf{M}\mathbf{U})diag(\gamma'^{-1})) \\ &= \sum_i \frac{\|(\mathbf{U}^T\mathbf{M}\mathbf{U})_{i.}\|_2^2}{\gamma'_i} \\ &= \sum_i \frac{\|(\mathbf{U}^T\mathbf{M}\mathbf{U})_{i.}\|_2^2}{\gamma_i} \text{ as } \gamma_i = 0 \text{ iff } (\mathbf{U}^T\mathbf{M}\mathbf{U})_i = 0 \\ &= \|\mathbf{U}^T\mathbf{M}\mathbf{U}\|_{group}^2 \end{aligned}$$

Hence, we show that

$$|tr(\mathbf{M})|^2 \leq \|\mathbf{U}^T\mathbf{M}\mathbf{U}\|_{group}^2.$$

$\mathbf{U}^T\mathbf{M}\mathbf{U}$ is a diagonal matrix of eigenvalues of $\mathbf{M}$ when the columns of $\mathbf{U}$ are the eigenvectors of $\mathbf{M}$. Moreover, the above inequality becomes equality when the columns of $\mathbf{U}$ are the eigenvectors of the positive semi-definite matrix $\mathbf{M}$. ∎

4.4 Sparse Recovery

In this section, we investigate some of the sparse recovery property of the trace norm peanlty. The discussion of our sparse recovery result is split into three parts. We first discuss the compatibility condition applicable to the trace norm penalty. We then state the consistency theory and finally present the proofs at the end of this section.

4.4.1 Compatibility condition

Suppose the true $\mathbf{M}^*$ has rank of r. And let B be the set of matrices where the eigenvectors corresponding to non-zero eigenvalues are orthogonal to space spanned by the eigenvectors corresponding to non-zero eigenvalues of $\mathbf{M}^*$

$$B = \{\mathbf{A} \in \mathbb{R}^{p\times p} | \text{eigen}(\mathbf{M}^*) \perp \text{eigen}(\mathbf{A})\}. \tag{4.2}$$

where eigen($\mathbf{A}$) represents the space spanned by the eigenvectors corresponding to non-zero eigenvalues.

Define an operation Π as the projection of a matrix to the set B. And let

$$\mathbf{A}'' = \Pi(\mathbf{A}),$$
$$\mathbf{A}' = \mathbf{A} - \mathbf{A}''.$$

This decomposition is very similar to the decomposition given in Section 3.5.1 for the group lasso case. We are now ready to state the compatibility condition.

Compatibility Condition There exists $\psi > 0$ such that for any symmetric matrix $\mathbf{A}$ with $||\mathbf{A}''||_* \leq 3||\mathbf{A}'||_*$,

$$||\mathbf{A}'||_*^2 \leq \frac{1}{16N\phi^2} \sum tr[\mathbf{A}(X_i - X_j)(X_i - X_j)^T]^2.$$

∎

Compared with the compatibility condition stated on page 15 for the entrywise lasso, $||\mathbf{A}''||_* \leq 3||\mathbf{A}'||_*$ is similar to the $\|\mathbf{A}_{S^c}\|_1 \leq 3\|\mathbf{A}_S\|_1$ condition and this allows a positive ϕ even when the set of observations does not span the space of symmetric matrices.

4.4.2 Interpretation of the compatibility condition

Since the trace norm $\|\mathbf{M}\|_*$ is invariant to orthogonal transformation, one can rotate the variables such that $\mathbf{M}^*$ becomes diagonal without affecting the trace norm. Therefore, without loss of generality, a rank r matrix $\mathbf{M}^*$ can be assumed to be diagonal with the first r diagonal entries non-zero and the rest of the entries zero. The corresponding eigenvectors of a diagonal $\mathbf{M}^*$ are the standard basis, for example,

$$v_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$v_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Under the standard basis, the set B as defined in (4.2) can be written as

$$B = \{\mathbf{A} \in \mathbb{R}^{p\times p} : \mathbf{A}_{ij} = 0 \text{ if } i \leq r \text{ or } j \leq r\}.$$

In other words, B is the set of matrices with non-zero only in the lower $(p-r)\times(p-r)$ block if $\mathbf{M}^*$ is diagonal with the first r diagonal entries non-zero. This decomposition is identical to the one given in Section 3.5.1 on page 35 for the group lasso case and hence most of the discussions of the compatibility condition of the group lasso penalty also apply to the trace norm penalty.

4.4.3 Consistency Theory

Assumptions 2.2.1 and 2.2.2 on the size of the error and independent structure are assumed. Recall that these assumptions are

Assumption 4.4.1 *We assume that the error is bounded, that is $|\varepsilon_{i,j}| \leq \delta^2$, and has mean zero.*

Assumption 4.4.2 *The errors $\varepsilon_{i,j}$ and $\varepsilon_{h,k}$ are independent if indices i, j, h, k do not overlap.*

Let $\hat{\mathbf{M}}$ minimizes the following loss function.

$$\hat{\mathbf{M}} = \text{argmin}_{\mathbf{M}\succeq 0} \sum_{ij} \frac{(d(i,j) - d_{\mathbf{M}}(i,j))^2}{N} + \lambda\|\mathbf{M}\|_*. \tag{4.3}$$

We now state the consistency theory for the trace norm penalty.

Theorem 4.4.3 *Let $\hat{\mathbf{M}}$ be the estimator in (4.3). Suppose that Assumptions 2.2.2 and 2.2.1 hold. Suppose that there exists some $c > 0$ such that $|X_{i,h}| \leq c$ for all $h = 1, \ldots, p$, and $i = 1, \ldots, n$. Let*

$$\lambda = 16c^2\delta^2 p\sqrt{\frac{t^2 + 4\log(p) + 2\log(n)}{(n-1)/2}}$$

Then, with probability at least $1 - 2\exp(-t^2/2)$,

$$\sum_{ij}(\hat{d}(i,j) - d^*(i,j))^2/N + \lambda||\hat{\mathbf{M}} - \mathbf{M}^*||_* \leq 4\frac{\lambda^2}{\phi^2}.$$

∎

The theory implies that the mean squared error converges at the rate of $O(p^2 \log p/n)$ which is slower than $O(p \log p/n)$ of group lasso and $O(\log p/n)$ of lasso.

If compatibility condition fails, a weaker version of consistency still holds.

Theorem 4.4.4 *Let $\hat{\mathbf{M}}$ be the estimator in (4.3). Suppose that Assumptions 2.2.2 and 2.2.1 hold. Suppose that there exists some $c > 0$ such that $|X_{i,h}| \leq c$ for all $h = 1, \ldots, p$, and $i = 1, \ldots, n$. Let*

$$\lambda = 16c^2\delta^2 p\sqrt{\frac{t^2 + 4\log(p) + 2\log(n)}{(n-1)/2}}$$

Then, with probability at least $1 - 2\exp(-t^2/2)$,

$$2\sum_{ij}(\hat{d}(i,j) - d^*(i,j))^2/N \leq 3\lambda||\mathbf{M}^*||_*$$

∎

This suggests that the mean squared error still converges at the slower rate of $O(\sqrt{p^2 \log p/n})$ if compatibility condition fails.

4.4.4 Proofs

The key of the proof depends on the following inequality which then allow us to bound the random noisy part of the problem.

Lemma 4.4.5

$$|\sum \epsilon_{i,j} tr((X_i - X_j)(X_i - X_j)^T(\hat{\mathbf{M}} - \mathbf{M}^*))/N| \leq ||\hat{\mathbf{M}} - \mathbf{M}^*||_*||\sum \varepsilon_{ij}(X_i - X_j)(X_i - X_j)^T/N||_{op}$$

where $||\mathbf{A}||_{op}$ denotes the operator norm, which equals the largest eigenvalues of $\mathbf{A}$.

We present two proofs of this fact. The first would be a simple direct proof and the second one would give more intuition of the problem.

Proof Let

$$\mathbf{A} := \hat{\mathbf{M}} - \mathbf{M}^*$$

$$\mathbf{B} := \sum \varepsilon_{ij}(X_i - X_j)(X_i - X_j)^T/N.$$

$\mathbf{B}$ can be eigendecomposed into $\mathbf{B} = \mathbf{V}\mathbf{D}\mathbf{V}^T$ where $\mathbf{D}$ is a diagonal matrix of eigenvalues of $\mathbf{B}$.

$$
\begin{aligned}
|tr(\mathbf{AB})| &= |tr(\mathbf{AVDV}^T)| \\
&= |tr(\mathbf{V}^T\mathbf{AVD})| \\
&\leq |tr(\mathbf{V}^T\mathbf{AV})|\,||\mathbf{D}||_\infty \\
&= |tr(\mathbf{A})|\,||\mathbf{B}||_{op} \\
&\leq ||\mathbf{A}||_*||\mathbf{B}||_{op}
\end{aligned}
$$

where we have used the following two facts. $\|\mathbf{D}\|_\infty$ equals the maximum entry of $\mathbf{D}$ which is the maximum eigenvalues of $\mathbf{B}$, hence, $\|\mathbf{D}\|_\infty = \|\mathbf{B}\|_{op}$. $tr(\mathbf{A}) \leq ||\mathbf{A}||_*$ because $tr(\mathbf{A})$ is the sum of eigenvalues of $\mathbf{A}$ and $||\mathbf{A}||_*$ is the sum of the absolute of the eigenvalues. ∎

The second proof would use the von Neumann trace inequality (Mirsky, 1975) and the Hölder's inequality. This proof gives more insight to the decomposition of singular values of the two matrices.

Proof Let $\sigma(\mathbf{A})$ denote the ordered singular values vector of $\mathbf{A}$ and $\sigma(\mathbf{B})$ denote the ordered singular values vector of $\mathbf{B}$.

Von Neumann trace inequality states that

$$|tr(\mathbf{AB})| \leq \langle \sigma(\mathbf{A}), \sigma(\mathbf{B}) \rangle$$

where $\langle ., . \rangle$ denotes the dot product between the two vectors.

Hölder's inequality states that

$$\langle \sigma(\mathbf{A}), \sigma(\mathbf{B}) \rangle \leq ||\sigma(\mathbf{A})||_1 ||\sigma(\mathbf{B})||_\infty.$$

The proof is completed by noting $||\sigma(\mathbf{A})||_1 = ||\mathbf{A}||_*$ and $||\sigma(\mathbf{B})||_\infty = ||\mathbf{B}||_{op}$. ∎

We now prove a bound of the empirical part $||\sum \varepsilon_{ij}(X_i - X_j)(X_i - X_j)^T/N||_{op}$.

Lemma 4.4.6

$$For\ \lambda_0 = 8c^2p\delta^2\sqrt{\frac{t^2 + 4\log(p) + 2\log(n)}{(n-1)/2}},$$

$$P(2||\sum_{ij} \varepsilon_{ij}(X_i - X_j)(X_i - X_j)^T||_{op}/N \leq \lambda_0) \geq 1 - \exp\left(-\frac{t^2}{2}\right)$$

Proof Let $S^{p-1} = \{u \in \mathbb{R}^p : ||u||_2 = 1\}$.

$$\begin{aligned}
||2\sum_{ij} \varepsilon_{ij}(X_i - X_j)(X_i - X_j)^T||_{op}/N &= \max_{v \in S^{p-1}} \frac{2\sum_{ij} \varepsilon_{ij} v^T(X_i - X_j)(X_i - X_j)^T v}{N} \\
&= \max_{v \in S^{p-1}} \frac{2\sum_{ij} tr(\varepsilon_{ij}(X_i - X_j)(X_i - X_j)^T vv^T)}{N} \\
&\leq \max_{v \in S^{p-1}} 8c^2\delta^2 \sqrt{\frac{t^2 + 4\log(p) + 2\log(n)}{(n-1)/2}} \|vv^T\|_1 \\
&\leq 8c^2 p\delta^2 \sqrt{\frac{t^2 + 4\log(p) + 2\log(n)}{(n-1)/2}}
\end{aligned}$$

In the last step, we applied the inequality $\|vv^T\|_1 \leq p\|vv^T\|_2 = p$ and the fact that $\|vv^T\|_2 = 1$. ∎

The remaining proofs then follow closely the one given in group lasso in the previous chapter, with the group penalty replaced by the trace norm penalty.

Lemma 4.4.7

$$\sum_{ij}(\hat{d}(i,j) - d^*(i,j))^2/N + \lambda||\hat{\mathbf{M}}||_* \leq 2\sum \varepsilon_{i,j} tr((X_i - X_j)(X_i - X_j)^T(\hat{\mathbf{M}} - \mathbf{M}^*))/N + \lambda||\mathbf{M}^*||_*$$

Proof Noting that

$$\sum_{ij}(d(i,j) - \hat{d}(i,j))^2/N + \lambda||\hat{\mathbf{M}}||_* \leq \sum_{ij}(d(i,j) - d^*(i,j))^2/N + \lambda||\mathbf{M}^*||_*,$$

The rest of the proof is identical to the one given for Lemma 2.5.2 on page 25 by rearranging the squared error terms. ∎

Lemma 4.4.8 *If* $\lambda \geq 4||\sum \varepsilon_{ij}(X_i - X_j)(X_i - X_j)^T||_{op}/N$, *then*

$$2\sum_{ij}(\hat{d}(i,j) - d^*(i,j))^2/N + 2\lambda||\hat{\mathbf{M}}||_* \leq \lambda\|\hat{\mathbf{M}} - \mathbf{M}^*\|_* + 2\lambda||\mathbf{M}^*||_*.$$

Proof By Lemma 4.4.7, then by Lemma 4.4.5,

$$2\sum_{ij}(\hat{d}(i,j)-d^*(i,j))^2/N+2\lambda||\hat{\mathbf{M}}||_* \leq 4\sum_{ij}\varepsilon_{i,j}tr((X_i-X_j)(X_i-X_j)^T(\hat{\mathbf{M}}-\mathbf{M}^*))/N + 2\lambda||\mathbf{M}^*||_*$$

$$2\sum_{ij}(\hat{d}(i,j)-d^*(i,j))^2/N+2\lambda||\hat{\mathbf{M}}||_* \leq 4\|\hat{\mathbf{M}}-\mathbf{M}^*\|_*\|\sum_{ij}\varepsilon_{ij}(X_i-X_j)(X_i-X_j)^T/N\|_{op} + 2\lambda||\mathbf{M}^*||_*$$

$$2\sum_{ij}(\hat{d}(i,j)-d^*(i,j))^2/N+2\lambda||\hat{\mathbf{M}}||_* \leq \lambda\|\hat{\mathbf{M}}-\mathbf{M}^*\|_*+2\lambda||\mathbf{M}^*||_*$$

∎

The following Lemma gives us the condition that compatibility condition is only required to hold over a cone.

Lemma 4.4.9 *Let* $\Delta=\hat{\mathbf{M}}-\mathbf{M}^*$*, if* $\lambda \geq 4||\sum\varepsilon_{ij}(X_i-X_j)(X_i-X_j)^T||_{op}/N$*, then* $||\Delta''||_* \leq 3||\Delta'||_*$.

Proof

$$\begin{aligned}||\hat{\mathbf{M}}||_* &= ||\hat{\mathbf{M}}+\Delta''+\Delta'||_* \\ &\geq ||\mathbf{M}^*+\Delta''||_*-||\Delta'||_* \quad \text{by triangle inequality} \\ &\geq ||\mathbf{M}^*||_*+||\Delta''||_*-||\Delta'||_*\end{aligned}$$

We applied the equality $||\mathbf{M}^*+\Delta''||_*=||\mathbf{M}^*||_*+||\Delta''||_*$ since $\mathbf{M}^*$ and Δ'' are orthogonal by construction.

By Lemma 4.4.8,

$$\begin{aligned}2\sum_{ij}(\hat{d}(i,j)-d^*(i,j))^2/N+2\lambda||\hat{\mathbf{M}}||_* &\leq \lambda\|\hat{\mathbf{M}}-\mathbf{M}^*\|_*+2\lambda||\mathbf{M}^*||_* \\ 0 &\leq \lambda/2||\hat{\mathbf{M}}-\mathbf{M}^*||_*+\lambda(||\mathbf{M}^*||_*-||\hat{\mathbf{M}}||_*) \\ &\leq \lambda/2||\Delta||_*+\lambda(||\Delta'||_*-||\Delta''||_*) \\ &= \lambda\left(\frac{3}{2}||\Delta'||_*-\frac{1}{2}||\Delta''||_*\right).\end{aligned}$$

∎

A very similar argument would also give the proof of theorem 4.4.4.

Proof of theorem 4.4.4 By Lemma 4.4.8,

$$2\sum_{ij}\frac{(d^*(i,j)-\hat{d}(i,j))^2}{N}+2\lambda||\hat{\mathbf{M}}||_*\leq\lambda\|\hat{\mathbf{M}}-\mathbf{M}^*\|_*+2\lambda||\mathbf{M}^*||_*$$

$$2\sum_{ij}\frac{(d^*(i,j)-\hat{d}(i,j))^2}{N}\leq\lambda\|\hat{\mathbf{M}}-\mathbf{M}^*\|_*+2\lambda\|\mathbf{M}^*\|_*-2\lambda\|\hat{\mathbf{M}}\|_*$$

$$2\sum_{ij}\frac{(d^*(i,j)-\hat{d}(i,j))^2}{N}\leq\lambda\|\hat{\mathbf{M}}\|_*+\lambda\|\mathbf{M}^*\|_*+2\lambda\|\mathbf{M}^*\|_*-2\lambda\|\hat{\mathbf{M}}\|_*$$

$$2\sum_{ij}\frac{(d^*(i,j)-\hat{d}(i,j))^2}{N}\leq3\lambda\|\mathbf{M}^*\|_*$$

∎

Lemma 4.4.10 *By picking* $\lambda\geq4||\sum\varepsilon_{ij}(X_i-X_j)(X_i-X_j)^T||_{op}$,

$$2\sum_{ij}(\hat{d}(i,j)-d^*(i,j))^2/N\leq3\lambda||\hat{\mathbf{M}}-\mathbf{M}^*||_*$$

Proof By Lemma 4.4.8,

$$2\sum_{ij}(\hat{d}(i,j)-d^*(i,j))^2/N+2\lambda||\hat{\mathbf{M}}||_*\leq\lambda||\hat{\mathbf{M}}-\mathbf{M}^*||_*+2\lambda||\mathbf{M}^*||_*\text{ by the choice of }\lambda$$

$$2\sum_{ij}(\hat{d}(i,j)-d^*(i,j))^2/N\leq\lambda||\hat{\mathbf{M}}-\mathbf{M}^*||_*+2\lambda(||\mathbf{M}^*||_*-||\hat{\mathbf{M}}||_*)$$

$$2\sum_{ij}(\hat{d}(i,j)-d^*(i,j))^2/N\leq\lambda||\hat{\mathbf{M}}-\mathbf{M}^*||_*+2\lambda(||\mathbf{M}^*-\hat{\mathbf{M}}||_*)\text{ by triangle inequality .}$$

∎

The following Lemma is the last step of our proofs.

Lemma 4.4.11 *By picking* $\lambda\geq4||\sum\varepsilon_{ij}(X_i-X_j)(X_i-X_j)^T||_{op}$,

$$\sum_{ij}(\hat{d}(i,j)-d^*(i,j))^2/N+\lambda||\hat{\mathbf{M}}-\mathbf{M}^*||_*\leq4\frac{\lambda^2}{\phi^2}$$

Proof By Lemma 4.4.10,

$$\begin{aligned}2\sum_{ij}(\hat{d}(i,j)-d^*(i,j))^2/N+\lambda||\hat{\mathbf{M}}-\mathbf{M}^*||_*&\leq4\lambda||\hat{\mathbf{M}}-\mathbf{M}^*||_*\\&\leq16\lambda\|(\hat{\mathbf{M}}-\mathbf{M}^*)'\|_*\\&\leq4\lambda\frac{\sqrt{\sum_{ij}(\hat{d}(i,j)-d^*(i,j))^2}}{\sqrt{N}\phi}\text{ compatibility condition}\\&\leq\sum_{ij}(\hat{d}(i,j)-d^*(i,j))^2/N+4\frac{\lambda^2}{\phi^2}.\end{aligned}$$

We applied the inequality $4ab \leq a^2 + 4b^2$ in the last step. ∎

The proof of the theorem is completed by combining the probability statement on $\lambda \geq 4||\sum \varepsilon_{ij}(X_i - X_j)(X_i - X_j)^T||_{op}$ in Lemma 4.4.6 and the above lemma.

4.5 Optimization with Trace norm penalties

The alternating direction method of multipliers introduced in Section 3.6 can be extended to handle the trace norm penalty. To solve the optimization problem with trace norm penalty

$$\text{argmin}_{\mathbf{M}} L(\mathbf{M}) + \lambda \|\mathbf{M}\|_*,$$

we introduce a new matrix variable $\mathbf{A}$ and transform the problem into

$$\text{argmin}_{\mathbf{M}} L(\mathbf{M}) + \lambda \|\mathbf{A}\|_* \text{ such that } \mathbf{M} - \mathbf{A} = 0.$$

The iterative updates are

$$\begin{aligned}
\mathbf{M}^{k+1} &:= \text{argmin}_{\mathbf{M}} L(\mathbf{M}) + \frac{\rho}{2}||\mathbf{M} - \mathbf{A}^k + u^k||_2^2, \\
\mathbf{A}^{k+1} &:= \text{argmin} \frac{\rho}{2}||\mathbf{M}^{k+1} + u^k - \mathbf{A}||_2^2 + \lambda \|\mathbf{A}\|_*, \\
u^{k+1} &:= u^k + \mathbf{M}^{k+1} - \mathbf{A}^{k+1}.
\end{aligned}$$

Assume that the loss function L is smooth, the computation of the $\mathbf{M}$ update and u update are straightforward. The term involving the trace norm is now within the $\mathbf{A}$ update step. The $\mathbf{A}$ update step can be computed by an eigen-decomposition of $\mathbf{M}^{k+1} + u^k$. An eigen-decomposition of $\mathbf{M}^{k+1} + u^k$ gives $\mathbf{M}^{k+1} + u^k = V \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p) V^T$ where V is the matrix of eigenvectors and λ the corresponding eigenvalues. The solution of the $\mathbf{A}$ update is simply soft-thresholding of the eigenvalues of $\mathbf{M}^{k+1} + u^k$

$$\mathbf{A}^{k+1} := V \text{diag}(S_{\lambda/\rho}(\lambda_1), S_{\lambda/\rho}(\lambda_2), \ldots, S_{\lambda/\rho}(\lambda_p)) V^T.$$

This can be derived by the fact that both the Frobenius norm and the trace norm are both invariant to orthogonal transformation.

$$\begin{aligned}
&\frac{\rho}{2}||\mathbf{M}^{k+1} + u^k - \mathbf{A}||_2^2 + \lambda \|\mathbf{A}\|_* \\
&= \frac{\rho}{2}||\mathbf{V}^T(\mathbf{M}^{k+1} + u^k)V - \mathbf{V}^T\mathbf{A}\mathbf{V}||_2^2 + \lambda \|\mathbf{V}^T\mathbf{A}\mathbf{V}\|_* \\
&= \frac{\rho}{2}||\text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p) - \mathbf{V}^T\mathbf{A}\mathbf{V}||_2^2 + \lambda \|\mathbf{V}^T\mathbf{A}\mathbf{V}\|_*
\end{aligned}$$

Hence, the minimum is attained when

$$\mathbf{V}^T\mathbf{A}\mathbf{V} = \text{diag}(S_{\lambda/\rho}(\lambda_1), S_{\lambda/\rho}(\lambda_2), \ldots, S_{\lambda/\rho}(\lambda_p)).$$

4.6 Ridge $\ell 2$ penalties

We have only covered $\ell 1$ type penalties. Another popular type of regularization would be ones that based on $\ell 2$ penalties. In ordinary least squares estimation, ridge regression (Hoerl and Kennard, 1970) minimizes the squared error loss with a bound on the $\ell 2$-norm of the coefficients. This effectively shrinks the estimates towards zero and prediction performance is improved by a suitable bias-variance tradeoff. There are many differences in the properties of the lasso and $\ell 2$ penalties, for example, ridge regression provides shrinkage but does not set any coefficients to zero as in $\ell 1$ penalization. In terms of predictive performance, Tibshirani (1996) and Fu (1998) revealed that neither forms of penalties dominate in a linear regression setting. Lasso performs better than ridge regression when the true model is sparse while ridge regression excels when there are large number of small effects.

4.6.1 Ridge matrix penalty

In the linear regression setting, the ridge regression is fitted by applying a regularization of the sum of the squared of the coefficients. We could generalise this to apply in the matrix setting. The $\ell 2$ extension of the entrywise $\ell 1$ norm $\|\mathbf{M}\|_1$ would be the squared entrywise $\ell 2$ norm

$$\|\mathbf{M}\|_2^2 = \sum_{i=1}^{p}\sum_{j=1}^{p}\mathbf{M}_{ij}^2.$$

which is the squared Frobenius norm. The $\ell 2$ extension of the trace penalty would be the squared $\ell 2$ norm of the singular values of $\mathbf{M}$ which equals to the trace of $\mathbf{M}^2$. A simple calculation shows that the trace of $\mathbf{M}^2$ equals to squared the Frobenius norm

$$\|\mathbf{M}\|_2^2 = \sum_{i=1}^{p}\sum_{j=1}^{p}\mathbf{M}_{ij}^2 = tr(\mathbf{M}^2).$$

Hence, regularization using squared Frobenius norm is a natural $\ell 2$ extension for both the trace norm penalty $\|\mathbf{M}\|_*$ and the lasso penalty $\|\mathbf{M}\|_1$. By penalizing the sum of the squared of the coefficients and the squared $\ell 2$ norm of the singular values,

we expect that regularization using squared Frobenius norm would provide shrinkage in both the entrywise level and in the singular values of the fitted metric. However, unlike the lasso or the trace norm penalty, none of the entries or singular values would be exactly zero. In the next section, we will compare squared Frobenius norm (ridge) regularization with other regularization methods in simulated examples.

4.7 Numerical Examples

In this section, we compare different types of regularization in simulated examples and classification tasks.

The setup for the simulated examples is identical to Section 3.7 in Chapter 3.

1. Select parameters covariance matrix Σ, underlying metric $\mathbf{M}^*$, dimensionality p and noise level σ.
2. Generate independent p-dimensional $X_i \sim N(0, \Sigma), i = 1, \ldots, 300$.
3. Simulate 500 independent distances with the distribution

$$d(h) = (X_{i(h)} - X_{j(h)})^T \mathbf{M}^* (X_{i(h)} - X_{j(h)}) + \varepsilon_h, h = 1, \ldots, 500.$$

   where $\varepsilon_m \sim N(0, \sigma^2)$, $i(h), j(h)$ are drawn uniformly from any integer between 1 and 300.
4. A further 500 independent distances are simulated as a validation set to decide on the appropriate level of regularization. Finally, another 500 independent distances are generated to serve as a test set to compute an estimation of the mean squared error.
5. The model is fitted with the training data and the regularization parameter selected by the value that minimizes the squared error loss of the validation data.
6. The mean squared error of the test set is recorded.
7. Step 2-6 are repeated 50 times.

We have previously compared the entrywise lasso penalty $\|.\|_1$ with the group penalty $\|.\|_{group}$ in Chapter 3. We now present more comparison with the entrywise lasso

penalty $\|.\|_1$, the group penalty $\|.\|_{group}$, the trace norm penalty $\|.\|_*$, the squared Frobenius (ridge) penalty $\|.\|_2^2$, forward greedy group selection (in the group selection example) and fits without regularization.

Low rank matrix

In this example, the metric $\mathbf{M}^*$ is chosen to be a matrix with all its entries being 1. The below decomposition shows that it is a rank 1 matrix.

$$\mathbf{M}^* = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} (1\ 1\ 1 \dots 1)$$

Therefore, $\mathbf{M}^*$ is low rank, however, none of its entry is zero. The parameters for the simulation are chosen as covariance matrix $\Sigma = \mathbf{I}$, the dimension $p = 20$ and noise level $\sigma = 3$. The median and standard deviation of the mean squared errors from 50 runs are reported in the following table.

	Median Mean Squared Error	Standard Deviation
Lasso	13.086	1.464
Group	13.258	1.605
Trace norm	9.976	0.749
Ridge	13.078	1.456
Without regularization	13.157	1.566

As expected, the trace norm regularization has the best performance since it can take advantage of the low rank property of $\mathbf{M}^*$ while there are no huge differences between other methods.

In order to investigate the effect of correlation between variables, we now set the covariance matrix $\Sigma_{i,j} = 0.5^{|i-j|}$. The dimension and noise level remains at $p = 20$ and $\sigma = 3$ respectively. The median and standard deviation of the mean squared errors from 50 runs are reported in the following table.

	Median Mean Squared Error	Standard Deviation
Lasso	23.910	6.155
Group	18.296	3.252
Trace norm	9.730	0.604
Ridge	18.195	3.234
Without regularization	24.089	6.251

The error of all methods is higher than the previous simulation when there is no correlation between covariates. The trace norm remains the best type of regularization.

The group and ridge penalty is able to provide an edge over the lasso regularization. The result for the ridge regularization might draw some similarity with the linear regression setting where Tibshirani (1996) demonstrated that ridge regression performs best when there is large number of small coefficients.

Full rank but sparse matrix

In this example, the metric $\mathbf{M}^*$ is chosen to be the identity matrix $\mathbf{I}$. $\mathbf{M}^*$ being an identity matrix has plenty of entrywise sparsity to exploit but the matrix is full rank. The parameters for the simulation are chosen as covariance matrix $\Sigma = \mathbf{I}$, the dimension $p = 20$ and noise level $\sigma = 3$. The dimension and noise level is chosen as $p = 20$ and $\sigma = 3$ respectively. The median and standard deviation of the mean squared errors from 50 runs are reported in the following table.

	Median Mean Squared Error	Standard Deviation
Lasso	10.012	0.677
Group	17.920	2.127
Trace norm	19.134	2.294
Ridge	17.125	2.333
Without regularization	19.125	2.347

Lasso has the much better accuracy than other method since it can take advantage of the entrywise sparsity of $\mathbf{M}^*$.

To investigate the effect of correlation between variables, we now set the covariance matrix to be $\Sigma_{i,j} = 0.5^{|i-j|}$. The dimension and noise level remains at $p = 20$ and $\sigma = 3$ respectively. The median and standard deviation of the mean squared errors from 50 runs are reported in the following table.

	Median Mean Squared Error	Standard Deviation
Lasso	10.636	1.169
Group	36.646	10.871
Trace norm	34.933	6.200
Ridge	30.741	6.214
Without regularization	35.517	7.552

Lasso continues to be the best performing type of regularization while other methods suffer a steep decline in accuracy. Ridge regularization appears to be performing slightly better than other remaining types of regularization.

Group sparsity

Group sparsity can be seen as special of both the low rank assumption and entrywise sparsity since a group sparse matrix with k non-zero rows would have at most rank k and has at most $k \times k$ non-zero entries. Therefore, it would be interesting to compare these three types of regularization in a group sparse scenario. We also compare greedy forward selection. The forward group selection algorithm begins with a zero fitted metric $\mathbf{M} = \mathbf{0}$. At each step, a whole row is added to the metric. The row that minimizes the validation error is added to the model at each step. The above process is repeated until the validation error cannot improve further.

First group sparse example We set the metric to be

$$\mathbf{M}^* = \begin{pmatrix} \mathbf{I}_{12} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}.$$

$\mathbf{M}^*$ has the first 12×12 block is the identity matrix while the rest of the matrix is zero.

The covariance matrix is Σ with each entry $\Sigma_{i,j} = 0.5^{|i-j|}$. The dimension and noise level is at $p = 20$ and $\sigma = 3$ respectively. The median and standard deviation of the mean squared errors from 50 runs are reported in the following table.

	Median Mean Squared Error	Standard Deviation
Lasso	9.998	0.703
Group	19.867	3.172
Trace norm	23.012	3.020
Ridge	21.814	2.875
Forward Selection	13.621	1.814
Without regularization	27.362	4.555

The regularization method listed in increasing order of error is lasso, forward selection, group lasso, ridge and trace norm.

Second group sparse example We set the metric to be

$$\mathbf{M}^* = \begin{pmatrix} \mathbf{I}_5 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}.$$

$\mathbf{M}^*$ has the first 5×5 block is the identity matrix while the rest of the matrix is zero.

The covariance matrix is Σ with each entry $\Sigma_{i,j} = 0.5^{|i-j|}$. The dimension and noise level is at $p = 20$ and $\sigma = 3$ respectively. The median and standard deviation of the mean squared errors from 50 runs are reported in the following table.

	Median Mean Squared Error	Standard Deviation
Lasso	9.405	0.715
Group	11.403	1.167
Trace norm	14.175	1.757
Ridge	15.506	2.017
Forward Selection	9.557	0.810
Without regularization	25.245	8.078

The regularization method listed in increasing order of error is lasso, forward selection, group lasso, trace norm and ridge. With higher degree of group sparsity, the forward selection is as competitive as the lasso penalty.

Third group sparse example We set the metric $\mathbf{M}^*$ to be

$$\mathbf{M}^* = \begin{pmatrix} 1 & 1 & \dots & 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & 1 & \dots & 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 1 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix}.$$

$\mathbf{M}^*$ has the first 12×12 block is the matrix of 1's while the rest of the matrix is zero. This matrix has rank 1 and it would be interesting to observe whether the trace norm penalty is able to take advantage of this fact.

	Median Mean Squared Error	Standard Deviation
Lasso	12.831	1.501
Group	12.894	1.500
Trace norm	10.136	0.849
Ridge	19.823	3.029
Forward Selection	11.692	1.374
Without regularization	25.944	7.177

The trace norm penalty has the best performance in this case when $\mathbf{M}^*$ is both group sparse and low rank. Forward selection has the second lowest error, beating both lasso and group lasso penalties.

Last group sparse example The last group sparse example is similar to the third example. The metric $\mathbf{M}^*$ has the first 5×5 block is the matrix of 1's while the rest of the matrix is zero. This matrix has rank 1 and it would be interesting to observe whether the trace norm penalty is able to take advantage of this fact.

The simulation setup is detailed as follows. The covariance matrix is $\Sigma_{i,j} = 0.5^{|i-j|}$. The dimension and noise level is at $p = 20$ and $\sigma = 3$ respectively. The median and standard deviation of the mean squared errors from 50 runs are reported in the following table.

	Median Mean Squared Error	Standard Deviation
Lasso	9.748	0.882
Group	10.062	1.026
Trace norm	9.988	0.961
Ridge	12.839	1.134
Forward Selection	9.381	0.714
Without regularization	13.247	1.107

The regularization method listed in increasing order of error is forward selection, lasso, trace norm, group lasso and ridge. Forward selection outperforms other regularization method and this agrees with our earlier findings in the second example that the forward selection performs well with higher degree of group sparsity. We also note that lasso becomes more accurate than trace norm regularization as the degree of group sparsity increases.

Summary of the simulation results

The trace norm, lasso and ridge penalty do not dominate each other. There are scenario in which one would prefer each type of penalty over the others:

1. Low rank but dense entrywise - trace norm regularization has the best performance, followed by ridge penalty, and trace norm penalty has the worst performance.

2. Full rank but sparse entrywise - lasso regularization is the best method and leads other methods by a large margin, followed by ridge penalty, and lasso penalty has the worst performance.

Ridge regularization can improve the predictive accuracy in all examples and the improvement is more noticeable when the covariates are correlated. However, trace norm and lasso regularization are more effective in exploiting the underlying sparsity than a ridge regularization can do. The group penalty cannot outperform the lasso in most scenarios, the only exception is when the matrix is dense entrywise and when the covariates are correlated.

For the group sparse scenarios,we found that the result depends on the sparsity level. With moderate number of non-zero rows, lasso is often the most effective. If the group sparse matrix is also of low rank, then trace norm regularization can outperform lasso. However, with small number of non-zero rows, forward selection becomes as competitive as lasso.

4.7.1 Classification

We analyzed the datasets from the UCI Machine Learning Repository (Frank and Asuncion, 2010) which we first investigated in Section 2.3.2. We randomly split the data into a training set (80%) and a testing set (20%) and we repeat the simulations 10 times. The amount of regularization is determined by cross validation. The following is the prediction accuracy of the test set.

	Lasso	Trace norm
Zoo	96.12(4.86)	95.26(5.23)
Vehicle	57.40(10.49)	56.90(10.20)
Vowel	92.22(2.37)	94.06(1.97)
Soybean	90.93(3.92)	91.22(1.44)

When compared to the result in Section 2.3.2 where the fitted $\hat{\mathbf{M}}$ is restricted to be diagonal, the differences in accuracies are small. Since the optimization with a diagonal $\hat{\mathbf{M}}$ is computationally much simpler, placing a diagonal restriction is a good compromise in practice.

4.8 Related Works

The distance metric learning problem can be seen as a special case for a more general regression problem of the form

$$y_i = tr(\mathbf{Z}_i \mathbf{M}^T) + \varepsilon_i \tag{4.4}$$

where ε_i is i.i.d. noise with mean zero, $\mathbf{Z}_i$ is a $p \times q$ data matrix. $\mathbf{M}$ is a $p \times q$ low rank coefficient matrix which we attempt to recover by the following optimization

$$\text{argmin}_{\mathbf{M}} \sum_{i=1}^{n} (y_i - tr(\mathbf{Z}_i \mathbf{M}^T))^2 + \lambda \|\mathbf{M}\|_*. \tag{4.5}$$

Regression of this form 4.5 has been studied in Negahban and Wainwright (2011). They proposed the restricted strong convexity condition as a condition for sparse recovery.

Restricted Strong Convexity There exists $\psi > 0$ such that for any symmetric matrix $\mathbf{A}$ with $||\mathbf{A}''||_* \leq 3||\mathbf{A}'||_*$,

$$||\mathbf{A}||_2^2 \leq \frac{1}{16N\phi^2} \sum tr[\mathbf{A}(X_i - X_j)(X_i - X_j)^T]^2.$$

The restricted strong convexity condition can be seen as analogous to the restricted eigenvalue condition (Bickel et al., 2009) in the regression setting. Restricted strong convexity has a close connection with the compatibility condition: using the inequality $||\mathbf{A}||_*^2 \leq p||\mathbf{A}||_2^2$, restricted strong convexity implies compatibility condition. Negahban and Wainwright (2011) showed that $\|\hat{\mathbf{M}} - \mathbf{M}^*\|_2^2$ converges at the rate of $O((p+q)/n)$ when the entries of Z_i are Gaussian distributed.

4.8.1 Matrix Completion

The matrix completion problem (Candès and Recht, 2009) can also be seen as a special case of 4.5. Suppose $\mathbf{C}^*$ is a low rank $p \times q$ matrix. The matrix completion problem is to investigate whether we can recover $\mathbf{C}^*$ by observing only a subset of entries of $\mathbf{C}^*$. Let O be the set of indices that $\mathbf{C}^*$ is observed. The set of observed entries has the following form

$$\mathbf{C}_{i,j} = \mathbf{C}^*_{i,j} + \varepsilon_{i,j} \quad \forall (i,j) \in O$$

where $\varepsilon_{i,j}$'s are i.i.d. mean zero noise bounded by a constant δ.

It can be rewritten as

$$\mathbf{C}_{i,j} = tr(\mathbf{J}_{i,j}(\mathbf{C}^*_{i,j})^T) + \varepsilon_{i,j} \quad \forall (i,j) \in O.$$

where $\mathbf{J}_{i,j}$ is the zero matrix with 1 at the $(i,j) - th$ entry. This shows the connection between matrix completion and the more general model presented in equation (4.4).

Suppose a proportion t of the entires are sampled and the samples are chosen randomly. Suppose further $\mathbf{C}^*$ has rank r, then by singular value decomposition,

$$\mathbf{C}^* = \sum_{k=1}^{r} \sigma_k u_k v_k^T$$

where σ's are the singular values, while u and v are the corresponding singular vectors. The condition for sparse recovery is that none of the singular vectors are spiky, i.e.

$$\|u_k\|_\infty \leq \sqrt{\mu/p} \text{ and } \|v_k\|_\infty \leq \sqrt{\mu/q},$$

for some constant μ of order $O(1)$. Candès and Plan (2010) showed that by solving

$$\hat{\mathbf{C}} = \text{argmin}_{\mathbf{X}} \|\mathbf{X}\|_* \text{ such that } |\mathbf{X}_{i,j} - \mathbf{C}_{i,j}| \leq \delta \quad \forall (i,j) \in O,$$

then

$$\|\hat{\mathbf{C}} - \mathbf{C}^*\|_2 \leq 4\sqrt{\frac{(2+t)\min(p,q)}{t}}\delta + 2\delta$$

with high probability.

4.8.2 Distance metric learning

The model we assumed for metric learning is the following

$$d(i,j) = (X_i - X_j)^T \mathbf{M}(X_i - X_j) + \varepsilon_{i,j}.$$

This can be rewritten into

$$\begin{aligned} d(i,j) &= tr[(X_i - X_j)\mathbf{M}(X_i - X_j)^T] + \varepsilon_{i,j} \\ &= tr[(X_i - X_j)(X_i - X_j)^T\mathbf{M}] + \varepsilon_{i,j}. \end{aligned}$$

Hence, distance metric learning can be seen as a special case of (4.5). The result we have proved in Section 4.4 shows that $\|\hat{\mathbf{M}} - \mathbf{M}^*\|_*^2$ converges at the rate of $O(p^2 \log p/n)$. Using the inequality $\|\hat{\mathbf{M}} - \mathbf{M}^*\|_2 \leq \|\hat{\mathbf{M}} - \mathbf{M}^*\|_*$, we derive a bound that $\|\hat{\mathbf{M}} - \mathbf{M}^*\|_2^2$ converges at the rate of $O(p^2 \log p/n)$. Given the similarities between the metric learning problem and the general regression problem studied in Negahban and Wainwright (2011), this leaves an open problem of whether it is possible to connect some of their theoretical results to establish a stronger bound of $O(p/n)$.

4.8.3 Further extension

Zou and Hastie (2005) proposed elastic net by combining both form of penalties. One explanation is that the lasso tends to select only one variable from the group when the a group of predictors are highly correlated, By augmenting the lasso $\ell 1$ penalty with the $\ell 2$ norm penalty, the elastic net is able to select a group of predictor simultaneously.

4.9 Johnson Lindenstrauss Lemma

A related result in finding a low rank projection for nearest neighbour is the Johnson Lindenstrauss Lemma (Johnson and Lindenstrauss, 1984). The theorem stated that a set of n arbitrary points can be embedded in a lower dimension with only small distortions of their pairwise Euclidean distances. We first state the Johnson-Lindenstrauss Lemma(Johnson and Lindenstrauss, 1984)

Theorem 4.9.1 *Let $\epsilon \in (0, 1/2)$ and let $x_1, \ldots x_n \in \mathbb{R}^K$ be arbitrary points. Let $m = O(\epsilon^{-2} log(n))$ be a natural number, then there exists a Lipschitz map $f : \mathbb{R}^K \to \mathbb{R}^m$ such that*

$$(1-\epsilon)||x_i - x_j||_2^2 \leq ||f(x_i) - f(x_j)||_2^2 \leq (1+\epsilon)||x_i - x_j||_2^2. \tag{4.6}$$

Dasgupta and Gupta (2003) show that a random projection into m dimensions given by Gaussian distribution would satisfy Johnson-Lindenstrauss Lemma with high probability, giving a short proof of the existence of a Johnson-Lindenstrauss embedding. There are other random matrices distribution that generates Johnson-Lindenstrauss embedding.

4.9.1 Proof of Johnson Lindenstrauss Lemma

Here, we outline a short probabilistic proof of this theorem given in Dasgupta and Gupta (2003). Let $\mathbf{A}$ be a $m \times p$ matrix with independent, identically distributed standard normal entries. Let the mapping function f be

$$f(x) = \frac{1}{\sqrt{k}}\mathbf{A}x.$$

The proof requires the following lemma which gives a tight concentration result for one observation.

Lemma 4.9.2 *For any p-dimensional vector x,*

$$P\left((1-\epsilon)\|x\|_2^2 \leq \left\|\frac{1}{\sqrt{m}}\mathbf{A}x\right\|_2^2 \leq (1+\epsilon)\|x\|_2^2\right) \geq 1 - 2\exp(-(\epsilon^2 - \epsilon^3)m/4) \tag{4.7}$$

Applying the above lemma with a union bound over the set of all pairwise distances would give the Johnson Lindenstrauss Lemma.

$$\begin{aligned}
&P((1-\epsilon)||x_i - x_j||_2^2 \leq ||f(x_i) - f(x_j)||_2^2 \leq (1+\epsilon)||x_i - x_j||_2^2 \quad \forall i,j = 1,\ldots,n) \\
\geq &1 - 2n^2 \exp(-(\epsilon^2 - \epsilon^3)m/4) \\
\geq &1 - \exp(-t^2)
\end{aligned}$$

by picking $m = \frac{4(8\log n + t^2)}{\epsilon^2}$.

The proof of Lemma 4.9.2 depends on the properties of normal and chi-squared distribution and the tight concentration property of a chi-squared variable.

Proof of Lemma 4.9.2 Since the sum of normal variables is also normally distributed, $\frac{\sum_{j=1}^p \mathbf{A}_{ij}x_j}{\|x\|_2} \sim N(0,1)$ for all $i = 1,\ldots,m$. The squared of a standard normal is chi-squared distributed with 1 degree of freedom, $\frac{(\sum_{j=1}^p \mathbf{A}_{ij}x_j)^2}{\|x\|_2^2} \sim \chi^2(1)$.

Expand $\|\mathbf{A}x\|_2^2$ to derive that

$$\frac{\|\mathbf{A}x\|_2^2}{\|x\|_2^2} = \frac{\sum_{i=1}^m (\sum_{j=1}^p \mathbf{A}_{ij}x_j)^2}{\|x\|_2^2} \sim \chi^2(m).$$

Hence, the statement in (4.7) is converted to proving a statement on the tight concentration of a chi-squared variable.

$$P((1-\epsilon) \leq \frac{1}{m}\chi^2(m) \leq (1+\epsilon)) \geq 1 - 2\exp(-(\epsilon^2 - \epsilon^3)m/4)$$

This can be proved by the Markov inequality.

$$\begin{aligned}
P(\chi^2(m) \geq (1+\epsilon)m) &= P(e^{\lambda\chi^2(m)} \geq e^{(1+\epsilon)m\lambda}) \\
&\leq \frac{E(e^{\lambda\chi^2(m)})}{e^{(1+\epsilon)m\lambda}} \\
&= e^{-(1+\epsilon)k\lambda}\left(\frac{1}{1-2\lambda}\right)^{\frac{k}{2}}
\end{aligned}$$

By the choice of $\lambda = \frac{e}{2(1+e)}$ and the inequality $1 + \epsilon \leq \exp(\epsilon - (\epsilon^2 - \epsilon^3)/2)$,

$$\begin{aligned}
P(\chi^2(m) \geq (1+\epsilon)m) &\leq ((1+\epsilon)e^{-\epsilon})^{k/2} \\
&\leq \exp\left(-\frac{k}{4}(\epsilon^2 - \epsilon^3)\right)
\end{aligned}$$

The other side of the inequality is proved similarly.

In addition to random projection given by Gaussian matrix, Krahmer and Ward (2011) showed that a random matrix satisfying Restricted Isometry Property (Candès and Tao, 2005) would also provide a Johnson-Lindenstrauss embeddings.

4.9.2 Discusion of Johnson Lindenstrauss Lemma

The result of Johnson Lindenstrauss theorem may at first seems surprising. The above probabilistic proof provides an explanation that it is closely related to the fact that an identity matrix can be well approximated by a low rank matrix $\mathbf{A}^T\mathbf{A}/k$ as $E(\mathbf{A}^T\mathbf{A}/k) = \mathbf{I}$. This result has seen application in speeding up the search for nearest neighbour, for example approximate nearest neighbour (Indyk and Motwani, 1998) and locality sensitive hashing (Datar and Indyk, 2004).

4.9.3 Johnson Lindenstrauss Lemma and distance metric learning

As a consequence of the Johnson Lindenstrauss Lemma, for any distance metric $\mathbf{M}$ and $0 < \epsilon < 0.5$, there exists another distance metric $\mathbf{M}'$ of rank $O(\log n/\epsilon^2)$ such that

$$(1-\epsilon)||(x_i-x_j)^T\mathbf{M}(x_i-x_j)||_2^2 \leq ||(x_i-x_j)^T\mathbf{M}'(x_i-x_j)||_2^2 \leq (1+\epsilon)||(x_i-x_j)^T\mathbf{M}(x_i-x_j)||_2^2. \tag{4.8}$$

By Cholesky decomposition, we obtain a matrix $\mathbf{C}$ with $\mathbf{M} = \mathbf{C}^T\mathbf{C}$. Applying a random projection in addition to the transformation given by $\mathbf{M}$ is equivalent to first rotate the data with the linear transformation matrix $\mathbf{C}$ and then project the data with the random projection matrix $\sqrt{1/k}\mathbf{A}$. Hence, the distance metric $\mathbf{M}'$ has the following form: $\mathbf{M}' = \mathbf{C}^T\mathbf{A}^T\mathbf{A}\mathbf{C}/k$. It might at first seem contradictory to have a consistency result that $\|\hat{\mathbf{M}} - \mathbf{M}^*\|_*$ converges and there exists some arbitrary metric $\hat{\mathbf{M}}'$ generated by random projection that well approximates the distances under metric $\hat{\mathbf{M}}$. This can be explained by the fact that $E(\mathbf{A}^T\mathbf{A}/k) = \mathbf{I}$ and hence one would expect that $\hat{\mathbf{M}}'$ would be concentrated around the original matrix $\hat{\mathbf{M}}$.

4.10 Discussion

A low rank solution can be found by enforcing a trace norm regularization. We connected the trace norm regularization with the group penalty introduced in Chapter 3. We derived the trace norm penalty as the group penalty after an orthogonal transformation. The optimization algorithm proposed in Chapter 3 was extended to cover the trace norm penalty. Results from simulations demonstrated that there is an

advantage of using trace norm regularization if $\mathbf{M}$ is of low rank. Although in various classification tasks, we noted that learning the a diagonal metric is often sufficient for a good classifier. Since the optimization with a diagonal $\hat{\mathbf{M}}$ is computationally much simpler, placing a diagonal restriction is a good compromise in practice.

Chapter 5

Logistic Loss with Rule Ensemble

We have formulated the metric learning problem into a regression problem in the previous chapter. One interpretation of the fitted quantities is that they represent the probability whether a pair belong to different classes. Therefore, it would be natural to use logistic loss for this type of problem. We formulate this idea into a logistic regression problem: the probability that two observations are of the same class diminishes when the distance between them is large. Imposing regularization allows one to potentially consider a very large number of covariates. We follow the proposal of Friedman and Popescu (2008) and expand the set of covariates to the set of rule ensembles.

Rules are hyper-rectangles defined in the design space. They can be expressed as simple statements on the input variable values and can be interpreted easily. Regression using rule ensembles was first suggested by Friedman and Popescu (2008). The distance between two observations depends on the number of rules they differ. By applying a suitable regularization, the number of rules picked is small and the results can be interpreted easily. The minimization problem is convex and can be solved by a modification of the coordinate descent algorithm proposed by Friedman et al. (2009).

This chapter is organized as follows. Our formulation of the distance metric learning problem using rule ensembles are presented in Section 5.1. Section 5.2 discusses the choices of tuning parameters and how the learned metric can be interpreted. The coordinate descent algorithm is detailed in Section 5.3. Several data analysis and performance comparison with other methods are given in Section 5.4. We conclude with a discussion in Section 5.5.

5.1 Metric learning using Rule Ensembles

Friedman and Popescu (2008) defined rule as a rectangular subspace of the p-dimensional space of the variables. A rule is defined as a rectangular subspace

$$Q_m = \{x = (x_{(1)} \dots x_{(p)}) : x_k \in I_k \text{ for } k = 1, \dots, p\},$$

where I_k is an interval defined on the support of the k$^{\text{th}}$ variable.

The indicator function for the rule m would be

$$r_m(x) = \begin{cases} 1 & \text{if } x \in Q_m \\ 0 & \text{otherwise} \end{cases} .$$

Rules can be expressed as simple statements. For example, in a spam email detection application, a rule could be *the percentage of the word "money"* ≤ 0.035 *and the percentage of the word "remove"* ≤ 0.08.

One computationally efficient way to obtain rules is by recursive tree algorithm. Random Forest(Breiman, 2001) is currently used to generate the ensemble of rules. By subsampling and choosing variables randomly, we can obtain a good ensemble of rules.

We restrict the number of variables that define a rule so that the result is more interpretable. Rule ensemble consists of rules defined by two variables often gives a good balance between the prediction accuracy and model complexity.

Friedman and Popescu (2008) considered the regression problem where the fitted value has the form

$$F(x) = \beta_0 + \sum_m \beta_m r_m(x).$$

Given observed value y_i, the aim is to minimize the loss

$$\text{argmin}_\beta \sum_i L(y_i, \beta_0 + \sum_m \beta_m r_m(x)) + \lambda \sum_{m=1}^{M} |\beta_m|$$

where L is a loss function and λ is a regularization parameter.

The amount of regularization applies to β_m depends on the scaling on $r_m(x)$. A common practice is to scale the covariates such that all covariates have unit variance. The transformation for the rule would be $r_m(x) \leftarrow r_m(x)/\sqrt{s_m(1-s_m)}$, where $s_m = \frac{1}{n}\sum_i r_m(x_i)$.

However, Friedman and Popescu (2008) used unnormalized rules which place a heavier penalty on rules that has sparse support $s_m \sim 0$ or dense support $s_m \sim 1$. This has the effect of reducing the variance of the fitted model since rules with small support or the complement of large support are defined by a smaller number of observations. When comparing the importance of different rules, they proposed the standardized predictor as a measure of relevance

$$I_m = |\beta_m| \sqrt{s_m(1 - s_m)}. \tag{5.1}$$

The corresponding local importance measure would be

$$I_m(x) = |\beta_m||r_m(x) - s_m|. \tag{5.2}$$

We will derive a similar importance measure in Section 5.2.2.

5.1.1 Using rules as the basis for distance metric learning

We now present our formulation of the metric learning problem in the framework set up in Section 1.1.

Suppose we have generated a set of M rules. Each observation x is mapped into M indicator variables. Let $\phi(x)$ be the vector function $\mathbb{R}^p \to \{1, 0\}^M$

$$\phi(x) := \begin{pmatrix} r_1(x) \\ r_2(x) \\ \vdots \\ r_M(x) \end{pmatrix}.$$

The distance would then be defined in terms of $\phi(x)$

Let $\beta_m \geq 0$ specifies the weights assigned to each rule. The distance between two observations is defined as

$$D_{i,j}(\beta) = \sum_{m=1}^{M} \beta_m |r_m(x_i) - r_m(x_j)|. \tag{5.3}$$

Therefore, we are interested in learning a set of non-negative coefficients β_m.

5.1.2 Generalized Linear model formulation

We now formulate our metric learning problem as a logistic regression.

Let

$$y_{i,j} = \begin{cases} 1 & \text{if observations i and j belong to different classes} \\ 0 & \text{otherwise} \end{cases}.$$

Let the probability of observing a pair of observations in different class be

$$P(y_{i,j} = 1|.) = \frac{1}{1 + e^{-(\beta_0 + D_{i,j})}}. \tag{5.4}$$

Similarly, we define the probability of a pair of observations in the same class

$$\begin{aligned} P(y_{i,j} = 0|.) &= \frac{1}{1 + e^{(\beta_0 + D_{i,j})}} \\ &= 1 - P(y_{i,j} = 1|.). \end{aligned} \tag{5.5}$$

Hence, the probability that observations i and j are of the same class decreases as the distance $D_{i,j}$ increases.

The loss function would be

$$l(\beta) = \frac{1}{N} \sum_{i \neq j} [y_{i,j} \log p_{i,j} + (1 - y_{i,j}) \log(1 - p_{i,j})] \tag{5.6}$$

where $p_{i,j} = P(y_{i,j} = 1|.)$ and N =the number of i, j pairs. This loss function resembles the binomial log likelihood. However, we should note that the $l(\beta)$ is not a log likelihood function, because the pairs $y_{i,j}$ are not independent and the set of marginals (5.4) do not define a valid joint distribution of all pairs $y_{i,j}$.

Our problem would be to minimize the negative "log likelihood" with an $\ell 1$ penalty

$$\min_{\beta} f_0(\beta) = \min_{\beta} -l(\beta) + \lambda \sum_{m=1}^{M} |\beta_m| \tag{5.7}$$

subject to the constraints

$$\beta_m \geq 0 \text{ for } m = 1, \ldots, M. \tag{5.8}$$

The $\ell 1$ regularization gives sparsity in rule selection. The positivity constraints (5.8) of β are imposed to ensure the matrix $\mathbf{M}$ is positive semi-definite. The positivity constraints also imply that we are finding a set of weights on the rules such that

observations of different classes are separated by the selected rules while each rule contains mostly observations of the same class.

The obtained distance metric is easy to interpret: the distance between two points is simply the sum of the weights β_m of the rules which they differ.

5.2 Interpretation of results

5.2.1 Choices of regularization parameters

The parameter λ determines the amount of regularization and hence the number of rules with non-zero weights. A smaller value of λ provides less regularization and results in larger number of non-zero coefficients. Our coordinate descent algorithm gives the fitted value β for a sequence of λ. This provides the necessary information to choose the regularization parameter λ by cross validation without refitting the model for different value of λ.

The scaling of $r_m(x)$ influences the amount of regularization applied to the weights β_m. A common approach is to normalise the variables such that all variables have unit variance. Let $s_m = \frac{1}{n}\sum_{i=1}^{n} r_m(x_i)$. The number of pairs with $|r_m(x_i) - r_m(x_j)| = 1$ would be $s_m(1-s_m)n^2$ and the total number of pairs is approximately $n^2/2$. Hence, the variance of $|r_m(x_i) - r_m(x_j)|$ would be roughly $2s_m(1-s_m)(1-2s_m(1-s_m))$. Each rule would be scaled $r_m(x) \leftarrow r_m(x)/t_m$ where $t_m = \frac{1}{\sqrt{2s_m(1-s_m)(1-2s_m(1-s_m))}}$.

We followed the suggestions given in Friedman and Popescu (2008) by using unnormalized rules. This choice prefers rules with approximately half the number of observations which has the effect of reducing the variance since the number of training observation pairs is larger for rules with approximately half the number of observations.

5.2.2 Rules importance

The weights β_m assigned to each rule can be seen as an importance measure of the rules. Here, we modified the two importance measures proposed by Friedman and Popescu (2008).

Friedman and Popescu (2008) suggested instead of the coefficient β, we should consider the coefficient of the standardized predictor as a measure of rule importance. We define the influence of a rule as

$$I_m = |\widehat{\beta_m}|\sqrt{2s_m(1-s_m)(1-2s_m(1-s_m))}$$

where $s_m = \frac{1}{n}\sum_{i=1}^{n} r_m(x_i)$.

For a specific pair of observations, we can define the corresponding the local measure of rule importance

$$I_m(x_i, x_j) = |\widehat{\beta_m}| \times \big||r_m(x_i) - r_m(x_j)| - 2s_m(1-s_m)\big|.$$

5.2.3 Input variable importance

We can mimic what have been done in random forest(Breiman, 2001) and use permutation of variable as a test of variable importance. Given a test set, we can permute the values of a particular variable and the decrease in prediction accuracy would indicate the importance of that particular variable in the chosen set of rules.

However, it should be interpreted carefully since the method tries to find the simplest interpretation of data. It might be true that effects of certain variables are masked by other variables and hence not included into the set of rules chosen.

5.2.4 Distance metric interpretation

Let $\mathbf{M}$ be the diagonal matrix with β on its diagonal entries. The distances can be written as

$$D_{ij} = (\phi(x_i) - \phi(x_j))^T \mathbf{M} (\phi(x_i) - \phi(x_j)).$$

Since $\mathbf{M}$ is a diagonal matrix with non-negative entries, $\mathbf{M}^{\frac{1}{2}}$ is well-defined,

$$D_{ij} = (\mathbf{M}^{\frac{1}{2}}\phi(x_i) - \mathbf{M}^{\frac{1}{2}}\phi(x_j))^T (\mathbf{M}^{\frac{1}{2}}\phi(x_i) - \mathbf{M}^{\frac{1}{2}}\phi(x_j))$$

The distance between x_i and x_j defined by our metric is equivalent to the Euclidean distance between $\mathbf{M}^{\frac{1}{2}}\phi(x_i)$ and $\mathbf{M}^{\frac{1}{2}}\phi(x_j)$.

Given the set of rules and its associated weights β_m, a new design matrix G of dimension $n \times M$ can be constructed. The i^{th} row of the matrix G is $\mathbf{M}^{\frac{1}{2}}\phi(x)$. The

pairwise distance measured by the Euclidean metric with the new design matrix G is identical to the distance obtained in the original problem.

5.2.5 Visualization

With the design matrix G described in the previous section, dimensionality reduction can be done on matrix G using Principal Components Analysis. The first few Principal Components of G would provide a useful low dimensional representation of the learned metric.

5.3 Pathwise coordinate optimization

Friedman et al. (2007) suggested that lasso optimization problem can often be solved quickly using coordinate descent with a decreasing schedule of the regularization parameter λ. Coordinate descent is attractive in solving the $\ell 1$ regularized problem because each update can be computed easily. Many iterations can be performed at a relatively low cost and still achieve high performance. Empirical result(Friedman et al., 2009) showed that the method is surprisingly fast and is competitive with other optimization algorithm such as the Least Angle Regression algorithm. The use of coordinate optimization to solve the LASSO problem was first presented by Fu (1998) as the shooting algorithm. Some recent papers applying this method includes, Friedman et al. (2009, 2008) in solving the graphical lasso and the regularized generalized linear models. Wu and Lange (2008) also presented coordinate descent algorithms to solve $\ell 1$ regularized problem.

5.3.0.1 Convergence properties

Bertsekas (1999) and Tseng (2001) proved that the limit points of coordinate descent algorithm converge to a minima of the objective function. However, there is little result regarding the rate of convergence of such methods. Saha and Tewari (2010) showed that cyclic coordinate minimization has at least a linear rate of convergence when a property called isotonicity holds. Isotonicity is a strong assumption: it requires the hessian of the objective function having no off-diagonal negative entries. This would require none of the covariates are positively correlated in the L1

regularized linear regression setting. Result on convergence rate can be obtained if the coordinate is updated randomly instead of in a cyclic order. Shalev-Shwartz and Tewari (2009) showed that the random coordinate descent does not require the isotonicity assumption and has a linear rate of convergence.

5.3.1 Algorithm

Our algorithm is essentially the "glmnet" algorithm presented in Friedman et al. (2009) with a modification to accommodate the positivity constraints (5.8). The algorithm consists of three main elements: pathwise optimization, quadratic approximation of the loss function (5.7) and coordinate-wise optimization.

5.3.1.1 Pathwise descent

The parameter λ controls the amount of regularization applied to the coefficients β. The regularization parameter λ naturally parametrises a solution path $\widehat{\beta}(\lambda)$. Least angle regression(Efron et al., 2004) computes the entire solution path for all value of λ for penalised least squares regressions. However, the solution path is not piecewise linear in the logistic regression case.

The strategy is to compute the solution of (5.7) for a decreasing sequence of λ, starting at the smallest value of λ such that $\beta_m(\lambda) = 0\ \forall m = 1, \ldots, M$. Denote this value of λ as $\lambda_{\max}$. We will derive an explicit formula for $\lambda_{\max}$ as a corollary of Lemma 5.3.1.

5.3.1.2 Quadratic approximation of the loss function

Generalized linear models are often fitted by iteratively reweighted least squares method (See for example McCullagh and Nelder, 1989). Each step is equivalent to solving the quadratic approximation of the likelihood. Motivated by the Iterative weighted least squares algorithm, the quadratic expansion of the objective function (5.7) around the current estimate $\tilde{\beta}$ would give the following approximation

$$\tilde{f}_0(\beta; \tilde{\beta}) = \frac{1}{2N} \sum_{i,j} \omega_{i,j}(z_{i,j} - \beta_0 - D_{i,j})^2 + \lambda \sum_{m=1}^{M} |\beta_m| + \text{constant}(\tilde{\beta}). \tag{5.9}$$

where

$$\omega_{i,j} = \tilde{p}_{i,j}(1-\tilde{p}_{i,j}) \text{ (working weights)},$$
$$z_{i,j} = \tilde{\beta}_0 + \tilde{D}_{i,j} + \frac{y_{i,j}-\tilde{p}_{i,j}}{\tilde{p}_{i,j}(1-\tilde{p}_{i,j})} \text{ (working values)},$$
$$\tilde{p}_{i,j} = \frac{1}{1+e^{-\tilde{\beta}_0-\tilde{D}_{i,j}}},$$
$$\tilde{D}_{i,j} = \sum_{m=1}^{M} \tilde{\beta}_m |r_m(x_i) - r_m(x_j)|.$$

5.3.1.3 Coordinate-wise optimization

A coordinate-wise update algorithm is performed to solve the approximation $\tilde{f}_0(\beta;\tilde{\beta})$:

1. For $m = 0, \ldots, M$, $\beta_m \leftarrow \text{argmin}_{\beta_m} \tilde{f}_0(\beta;\tilde{\beta})$.
2. Repeat step 1 until convergence.

We update each coefficient sequentially until convergence.

5.3.1.4 Combining three elements

The three ideas translated into three layer of loops. The outer loop generates a decreasing sequence of the regularization parameter λ. The middle loop updates the quadratic approximation (5.9). The inner loop performs optimization using coordinate descent. The initialization consists of finding the smallest regularization parameter λ such that the coefficients $\beta_m = 0 \forall m = 1, \ldots, M$. The algorithm is outlined below.

Algorithm	
Initialization:	Set $\beta_m = 0$ for $m = 1, \ldots, M$. Set β_0 such that $\frac{y_{ij}}{N} = \frac{1}{1+e^{-\beta_0}}$. Set $\lambda = \lambda_{\max}$.
Outer loop:	Decrement λ .
Middle loop:	Update the approximation $\tilde{f}_0(\beta;\tilde{\beta})$ around the current estimate $\tilde{\beta}$.
Inner loop:	Minimize $\tilde{f}_0(\beta;\tilde{\beta})$ using a coordinate descent algorithm.

5.3.2 Coordinate descent step

We now derive explicit formula for the coordinate descent step.

Differentiating equation (5.9) with respect to β_m,$m = 1, \ldots, M$, assume $\beta_m > 0$,

$$\frac{\partial \tilde{f}_0}{\partial \beta_m} = -\frac{1}{N} \sum_{i,j} \omega_{i,j}(z_{i,j} - \beta_0 - D_{i,j})|r_m(x_i) - r_m(x_j)| + \lambda \tag{5.10}$$

Equates to zero,

$$\frac{1}{N} \sum_{i,j} \omega_{i,j}(z_{i,j} - \beta_0 - D_{i,j})|r_m(x_i) - r_m(x_j)| - \lambda = 0.$$

Rearranging term, the minimum occurs when β_m equals

$$\alpha_m(\beta) := \frac{\sum_{i,j} \omega_{i,j}|r_m(x_i) - r_m(x_j)|(z_{i,j} - \beta_0 - \sum_{k=1,k\neq m}^{M} \beta_k|r_k(x_i) - r_k(x_j)|) - N\lambda}{\sum_{i,j} \omega_{i,j}|r_m(x_i) - r_m(x_j)|^2}.$$

Let S be the threshold function:

$$S(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}.$$

After imposing the positivity constraints(5.8), the coordinate-wise update is simply

$$\hat{\beta}_m = S(\alpha_m(\beta)) \text{ for } m = 1, \ldots, M. \tag{5.11}$$

The intercept is unconstrained, the coordinate-wise update would be

$$\alpha_0(\beta) = \frac{\sum_{i,j} \omega_{i,j}(z_{i,j} - \sum_{k=1}^{M} \beta_k|r_k(x_i) - r_k(x_j)|)}{\sum_{i,j} \omega_{i,j}}. \tag{5.12}$$

It is important to note that the coordinate-wise update (5.11) and (5.12) are continuous functions of β. These equations have a close connection with the Karush-Kuhn-Tucker conditions of the optimization problem (5.7).

5.3.3 Convergence property of the algorithm

In this section, we will prove that all the limit points of our coordinate descent algorithm converges to the minimum. The Karush-Kuhn-Tucker(Kuhn and Tucker, 1951) conditions of the optimization problem (5.7) and the continuity in the coordinate update function (5.11) are the two important properties used in our proof.

Lemma 5.3.1 *The following equations are both necessary and sufficient if β is an optimum point:*

$$\frac{1}{N}\sum_{i,j}(y_{i,j}-p_{i,j})|r_m(x_i)-r_m(x_j)|-\lambda\begin{cases}=0 & \text{if } \beta_m>0\\ \leq 0 & \text{otherwise}\end{cases} \quad \text{for } m=1,\ldots,M, \tag{5.13}$$

$$\frac{1}{N}\sum_{i,j}(y_{i,j}-p_{i,j})=0. \tag{5.14}$$

Proof Problem (5.7) is a problem with convex objective and affine boundaries. A version of the Slater's conditions(Slater, 1959) given in Boyd and Vandenberghe (Boyd and Vandenberghe, 2004, P.226) imply that strong duality holds for this problem. One consequence is that the Karush–Kuhn–Tucker conditions would be both necessary and sufficient at the optimal point.

We first consider the Lagrangian of the original problem:

$$-l(\beta)+\lambda\sum_{m=1}^{M}\beta_m-\sum_{m=1}^{M}\mu_m\beta_m$$

The Karush-Kuhn-Tucker conditions are

$$\beta_m \geq 0 \text{ for } m=1,\ldots,M, \tag{5.15}$$

$$\mu_m\beta_m=0 \text{ for } m=1,\ldots,M, \text{ (complementary slackness)} \tag{5.16}$$

$$\mu_m \geq 0 \text{ for } m=1,\ldots,M, \tag{5.17}$$

$$-\frac{\partial l(\beta)}{\partial\beta_m}+\lambda-\mu_m=0 \text{ for } m=1,\ldots,M, \tag{5.18}$$

$$\frac{\partial l(\beta)}{\partial\beta_0}=0. \tag{5.19}$$

Hence, β is a solution to problem (5.7) if and only if

$$-\frac{\partial l(\beta)}{\partial\beta_m}+\lambda=0 \text{ if } \beta_m>0 \text{ for } m=1,\ldots,M \tag{5.20}$$

$$-\frac{\partial l(\beta)}{\partial\beta_m}+\lambda\geq 0 \text{ if } \beta_m=0 \text{ for } m=1,\ldots,M, \tag{5.21}$$

$$\frac{\partial l(\beta)}{\partial\beta_0}=0. \tag{5.22}$$

It remains to show that

$$\frac{\partial l(\beta)}{\partial \beta_m} = \frac{1}{N}\sum_{i,j}(y_{i,j} - p_{i,j})|r_m(x_i) - r_m(x_j)| \text{ for } m = 1, \ldots, M, \tag{5.23}$$

by differentiating the log likelihood (5.6),

$$\begin{aligned}
\frac{\partial l(\beta)}{\partial \beta_m} &= \frac{1}{N}\sum_{i,j}\left[\frac{y_{i,j}}{p_{i,j}} - \frac{1-y_{i,j}}{1-p_{i,j}}\right]\frac{\partial p_{i,j}}{\partial \beta_m} \\
&= \frac{1}{N}\sum_{i,j}\left[\frac{y_{i,j} - p_{i,j}}{p_{i,j}(1-p_{i,j})}\right] p_{i,j}(1-p_{i,j})|r_m(x_i) - r_m(x_j)| \\
&= \frac{1}{N}\sum_{i,j}(y_{i,j} - p_{i,j})|r_m(x_i) - r_m(x_j)|.
\end{aligned}$$

Similarly, we can show that

$$\frac{\partial l(\beta)}{\partial \beta_0} = \frac{1}{N}\sum_{i,j}(y_{i,j} - p_{i,j}). \tag{5.24}$$

We obtain the result by substituting equations (5.23) and (5.24) into equations (5.20) to (5.22).

An immediate implication of the above result is the following corollary.

Corollary 5.3.2 *The smallest value of the regularization parameter λ that has coefficients $\beta_m = 0$ for all $m = 1, \ldots, M$ is $\lambda_{max} := \max_m \frac{1}{N}\sum_{i,j}(y_{i,j} - p_{i,j})|r_m(x_i) - r_m(x_j)|$, where $\beta_0 = -\log\left(\frac{N}{\sum_{i,j} y_{ij}} - 1\right)$ and $p_{i,j} = \frac{1}{1+e^{-\beta_0}}$.*

Proof The unconstrained term β_0 is chosen such that equation (5.14) is satisfied when other terms $\beta_m = 0$ for $m = 1, \ldots, M$.

$$\begin{aligned}
\frac{1}{N}\sum_{i,j}(y_{i,j} - p_{i,j}) &= \frac{1}{N}\sum_{i,j}(y_{i,j} - \frac{1}{1+e^{-\beta_0}}) \\
&= \frac{1}{N}\sum_{i,j}(y_{i,j} - \frac{y_{i,j}}{N}) \\
&= 0.
\end{aligned}$$

By Lemma 5.3.1, β_m is zero if and only if

$$\frac{1}{N}\sum_{i,j}(y_{i,j} - p_{i,j})|r_m(x_i) - r_m(x_j)| - \lambda \leq 0.$$

Hence,

$$\lambda_{\max} := \max_m \frac{1}{N} \sum_{i,j} (y_{i,j} - p_{i,j}) |r_m(x_i) - r_m(x_j)|$$

is the smallest value of the regularization parameter λ that has coefficients $\beta_m = 0$ for all $m = 1, \ldots, M$.

We now prove a result about limit points of our coordinate descent algorithm.

Proposition 5.3.3 *Every limit point of our cyclic coordinate descent algorithm converges to the solution of (5.7).*

Proof The key idea of this proof is to take limits of the coordinate update functions (5.11) and (5.12) and shows that they are equivalent to the conditions given in Lemma 5.3.1.

The functions (5.11) and (5.12) computing each update are continuous

$$\hat{\beta}_m = S(\alpha_m(\beta)),$$
$$\hat{\beta}_0 = \alpha_0(\beta).$$

Let the limit of β be β^*: $\lim_{k\to\infty} \beta^{(k)} = \beta^*$.

Hence, taking limit $k \to \infty$ of both sides,

$$\beta^*_m = S(\alpha_m(\beta^*)) \forall m = 1, \ldots, M, \tag{5.25}$$
$$\beta^*_0 = \alpha_0(\beta^*). \tag{5.26}$$

If $\beta^*_m > 0$, equation (5.25) implies that equation (5.10) holds:

$$\alpha_m(\beta^*) = \beta^*_m$$
$$\frac{1}{N} \sum_{i,j} \omega^*_{i,j} (z^*_{i,j} - \beta^*_0 - D^*_{i,j}) |r_m(x_i) - r_m(x_j)| - \lambda = 0.$$

Substitute in $z^*_{i,j} = \beta^*_0 + D^*_{i,j} + \frac{y_{i,j} - p^*_{i,j}}{p^*_{i,j}(1-p^*_{i,j})}$ and $\omega^*_{i,j} = p^*_{i,j}(1 - p^*_{i,j})$, we obtain

$$\frac{1}{N} \sum_{i,j} (y_{i,j} - p^*_{i,j}) |r_m(x_i) - r_m(x_j)| - \lambda = 0. \tag{5.27}$$

If $\beta_m^* = 0$, equation (5.25) implies that

$$\alpha_m(\beta^*) \leq 0$$
$$\frac{1}{N}\sum_{i,j}(y_{i,j} - p_{i,j}^*)|r_m(x_i) - r_m(x_j)| - \lambda \leq 0. \tag{5.28}$$

Similarly, equation (5.26) implies that

$$\frac{1}{N}\sum_{i,j}(y_{i,j} - p_{i,j}^*) = 0. \tag{5.29}$$

Equations (5.27), (5.28) and (5.29) are the conditions given in equation (5.13) and (5.14). Hence, every limit point of cyclic coordinate descent is a solution of (5.7).

The proof that a limit point exists is more technical. The main obstacle is that the full Newton step generated by middle loop might not be a descent step without step length selection. Friedman et al. (2009) noted that implementing a step length selection algorithm would be computationally costly and we have not experienced any non-convergence issues with the algorithm.

When β is sufficiently close to β^*, the middle loop would always be a descent step. The middle loop would then generate a sequence $\beta^{(k)}$ which $f_0(\beta^{(k)}) \geq f_0(\beta^{(k+1)})$. Assume further that the level set $\{\beta : f_0(\beta) \leq f_0(\beta^{(1)}))\}$ is compact, then by Bolzano-Weierstrass theorem, there exists a subsequence of $\beta^{(k)}$ that converges. The subsequence converges to β^* by proposition 5.3.3.

5.4 Data analysis

In this section, we apply our method to several datasets and compare its performance with other methods.

5.4.1 Wine dataset

This dataset is taken from the UCI Machine Learning Repository (Frank and Asuncion, 2010). It contains 13 variables and 178 observations. Each observation is labelled as one of the three types of wine.

This is an easy classification task where the noise is low and there is a clear separation of classes. The first two principal components of the original data(after rescaling and centering)(Figure 5.1) show clear separation of the three classes.

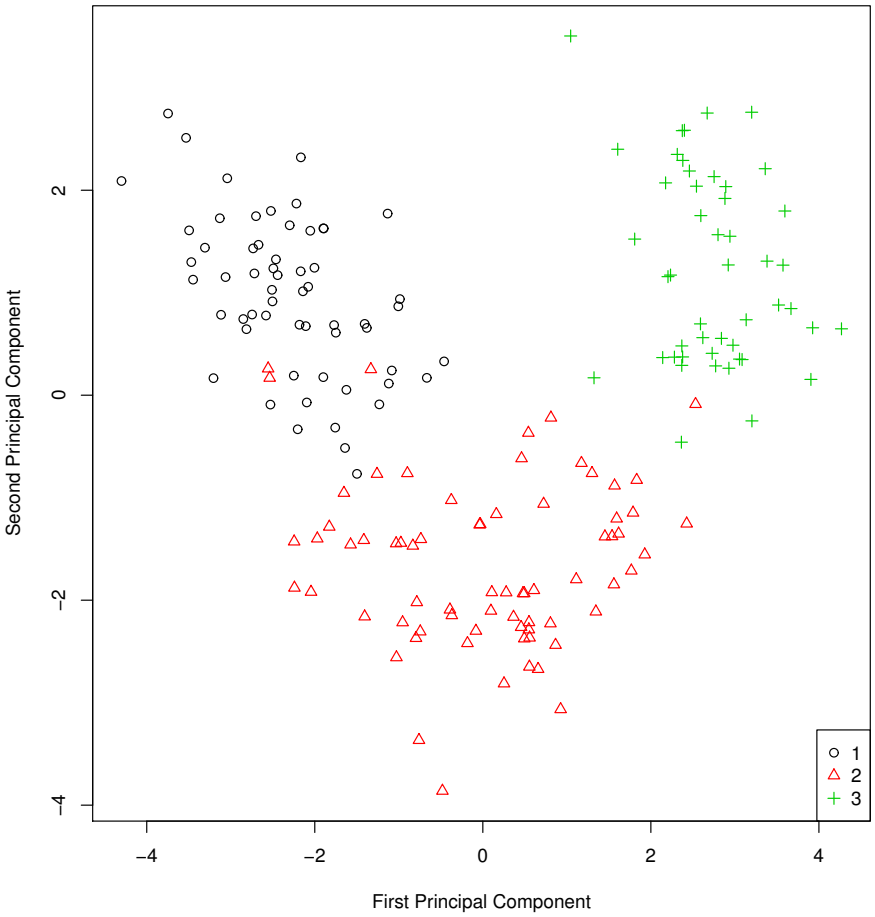


Figure 5.1: The first two Principal Components of the wine dataset

5.4.1.1 Choices of parameters

We now investigate the effect of the number of variables defining each rule and the effect of the size of the rule ensemble.

We perform a 5-fold cross validation to investigate the effect of the degree of interaction(number of variables defining each rule) on the prediction accuracy. The test accuracy reported is the average of 10 different runs of 5-fold cross validation:

Degree of interaction	1	2	3	4
Test Accuracy	62.7	93.8	93.8	94.4

There is a huge difference in predictive performance as the degree of interaction increases from 1 to 2. Setting the degree of interaction above 2 does very little to improve the predictive performance. Hence, setting degree of interaction to two seems to be a good balance between accuracy and rule complexity.

We now investigate the effect of the size of the rule ensembles on test accuracy by 5-fold cross validation. The test accuracy is again the average of 10 runs of 5-fold cross validation. The result is plotted in Figure 5.2.

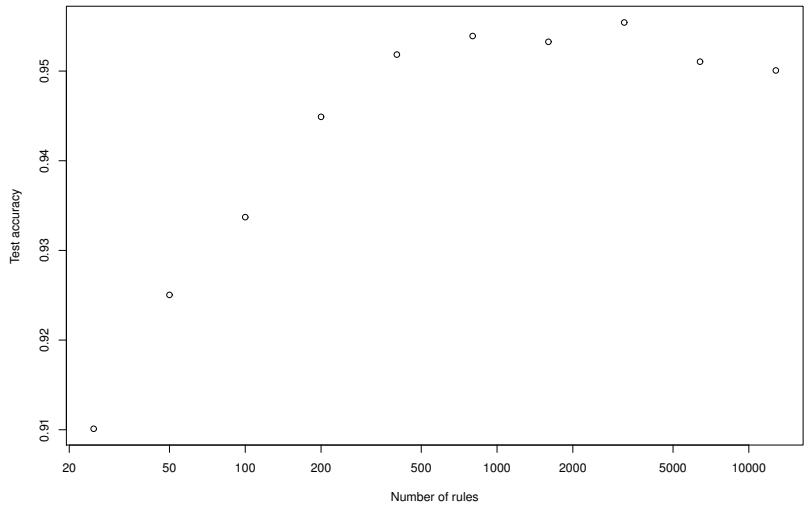


Figure 5.2: The effect of the number of rules in the initial ensembles on the prediction accuracy

Very small size of rule ensemble has an impact on prediction accuracy. The prediction accuracy seems to level off when the number of rules reaches 1000.

5.4.1.2 Rule metric

We illustrate our method in this section. The degree of interaction(number of variables define a rule) is set to two and the size of the rule ensemble is 1000. The regularization parameter λ is chosen by 10-fold validation. Figure 5.3 shows how the coefficients β vary as the regularization parameter λ decreases.

The learned rule metric consists of 49 rules with non-zero coefficients. We investigate the rules with highest influence:

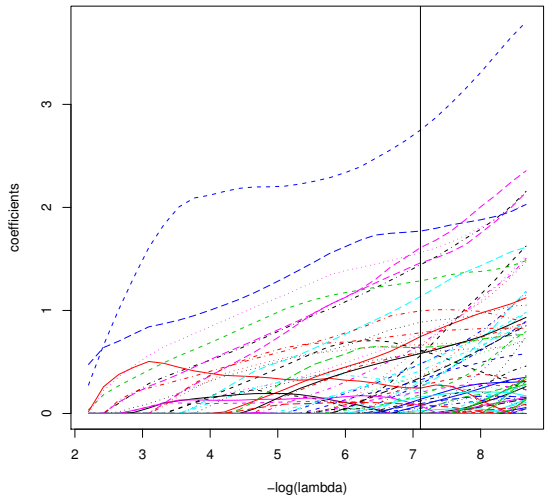


Figure 5.3: The coefficients β along the regularization path. The vertical line indicates the regularization parameter λ chosen by cross validation.

Class distribution(type 1/2/3) of the dataset:	59/71/48
Influence:	1.34
β:	2.75
Constraints:	Color intensity $>$ 3.65 and Flavanoids$\leq$ 1.425
Class distribution(type 1/2/3):	0/1/47
Influence:	0.882
β:	1.77
Constraints:	Alcohol$>$12.73 and Flavanoids$>$2.16
Class distribution(type 1/2/3):	59/4/0
Influence:	0.802
β:	1.61
Constraints:	0.745$<$Hue$\leq$1.295 and Proline$>$722.5
Class distribution(type 1/2/3):	58/5/0
Influence:	0.779
β:	1.56
Constraints:	Color intensity$>$3.82 and Ash$>$2.06
Class distribution(type 1/2/3):	54/7/48
Influence:	0.726
β:	1.46
Constraints:	Color intensity $>$3.49 and Proline $>$787.5
Class distribution(type 1/2/3):	54/0/5

The rules themselves are readily interpretable. For example, the first rule gives the constraints Color intensity $>$ 3.65 and Flavanoids$\leq$ 1.425, observations that satisfy the constraints consist of mainly observations of wine type 3. The distance between observations within the rule and observations that have Color intensity$\leq$ 3.65 or

Flavanoids > 1.425 would be at least 2.75 under the new metric.

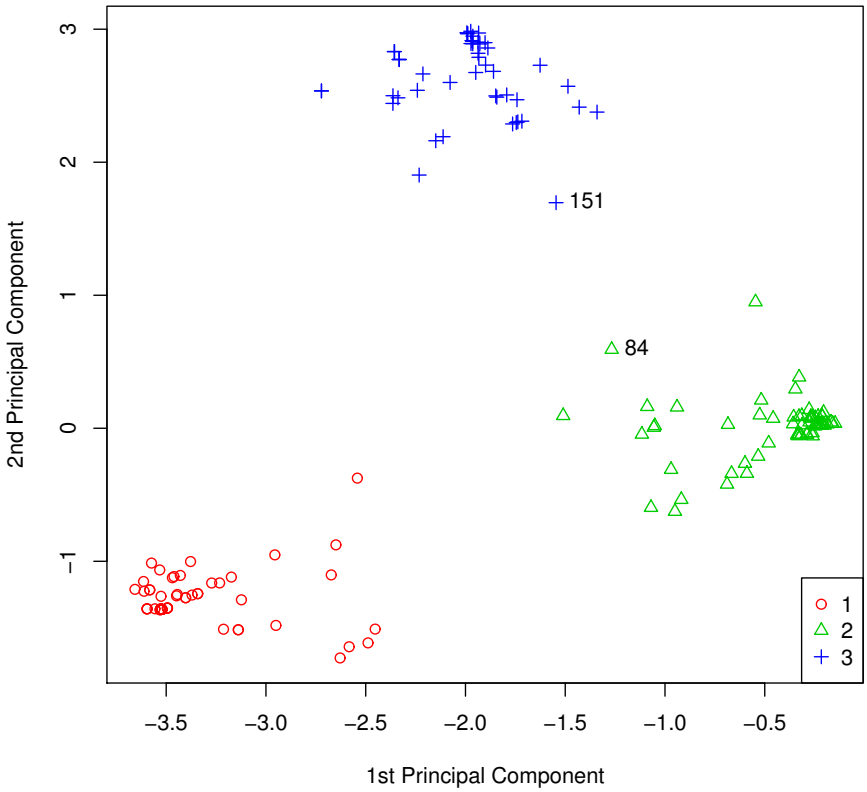


Figure 5.4: The 2D projection of the data using the rule metric by the first two Principal components of the matrix G

In Section 5.2.4, we presented a method to give a useful lower dimensional projection of the data. The first two principal components of the matrix G provides a useful 2-dimensional projection of the data as shown in Figure 5.4. The local influence of the two labelled input points are computed and the rules with highest local influences are:

Class distribution(type 1/2/3) of the dataset:	59/71/48
Local Influence:	1.67
β:	2.75
Constraints:	Color intensity>3.65 and Flavanoids $\leq$1.425
Class distribution(type 1/2/3):	0/1/47
Local Influence:	0.810
β:	1.77
Constraints:	Alcohol>12.73 and Flvanoids>2.16
Class distribution(type 1/2/3):	59/4/0
Local Influence:	0.751
β:	1.44
Constraints:	Proline>655 and Alcohol>13.02
Class distribution(type 1/2/3):	57/1/13
Local Influence:	0.741
β:	1.56
Constraints:	Color intensity>3.82 and Ash > 2.06
Class distribution(type 1/2/3):	54/7/48
Local Influence:	0.736
β:	1.61
Constraints:	0.745<Hue$\leq$1.295 and Proline>722.5
Class distribution(type 1/2/3):	58/5/0

The local influence measure gives us the set of rules that determines the distance between the two labelled wines.

5.4.1.3 Comparison with Linear discriminant analysis and classification tree

Linear Discriminant analysis gives two discriminant directions which provides meaningful lower dimensional representation of the data(Figure 5.5). The projection given by the linear discriminants is similar to the projection obtained by our metirc. The coefficients of the linear discriminants are

	LD1	LD2
Alcohol	-0.403399781	0.8717930699
‘Malic acid‘	0.165254596	0.3053797325
Ash	-0.369075256	2.3458497486
‘Alcalinity of ash‘	0.154797889	-0.1463807654
Magnesium	-0.002163496	-0.0004627565
‘Total phenols‘	0.618052068	-0.0322128171
Flavanoids	-1.661191235	-0.4919980543
‘Nonflavanoid phenols‘	-1.495818440	-1.6309537953

```
Proanthocyanins                0.134092628 -0.3070875776
'Color intensity'              0.355055710  0.2532306865
Hue                           -0.818036073 -1.5156344987
'OD280/OD315 of diluted wines' -1.157559376  0.0511839665
Proline                       -0.002691206  0.0028529846
```

which can be difficult to interpret.

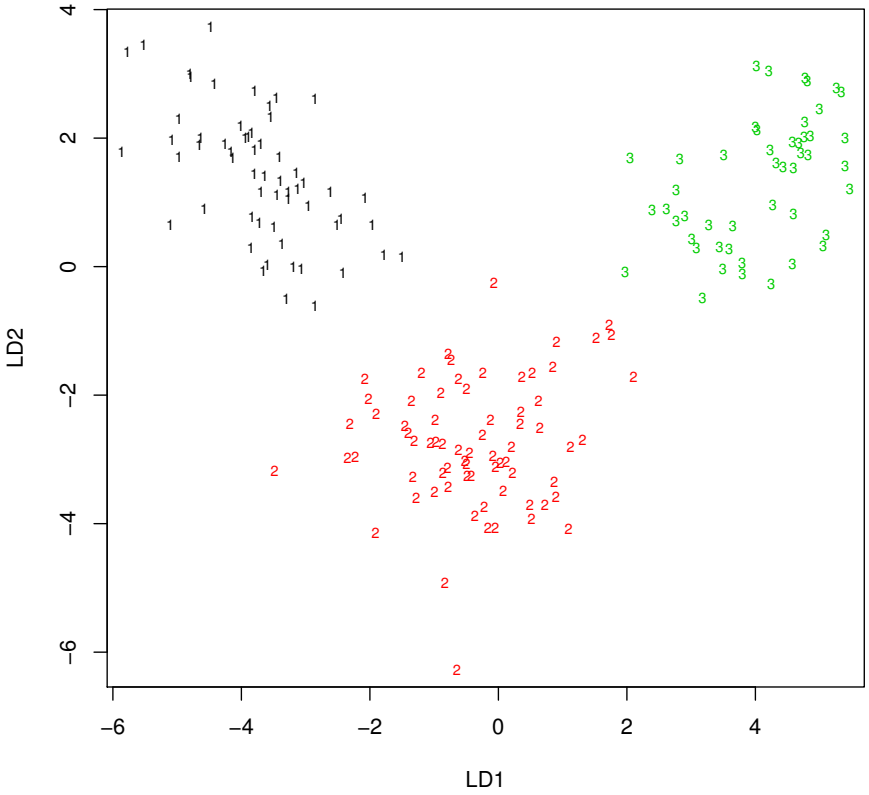


Figure 5.5: The projection of the data given by the two linear discriminants

Classification tree obtained by recursive partitioning gives a much simpler interpretation(Figure 5.6). The tree is pruned by the "+1 standard deviation" rule as suggested in Venables and Ripley (2002, Section 9.2). The tree is equivalent to the following rules:

- When Proline$\geq$ 755 and Flavanoids>= 2.165, then it is wine type 1.
- When Proline$\geq$ 755 and Flavanoids< 2.165, then it is wine type 3.

- When Proline< 755 and OD280/OD315 of diluted wines>= 2.1, then it is wine type 2.
- When Proline< 755 and OD280/OD315 of diluted wines< 2.1, then it is wine type 3.

The learned rule metric consists of 49 rules with non-zero coefficients is considerably more complex than the classification tree. However, imposing a tree structure incurs a penalty in predictive accuracy which we will see in Section 5.4.3.

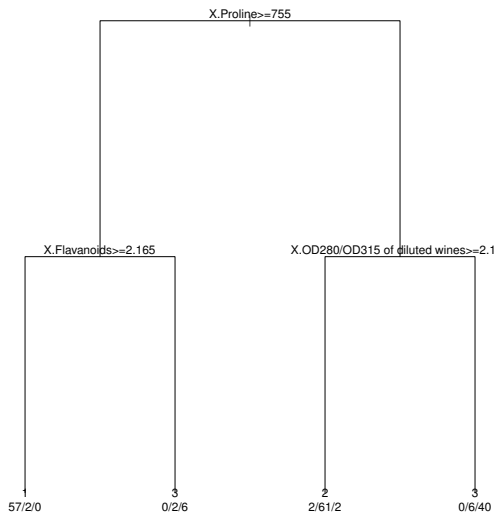


Figure 5.6: Classification tree of the wine dataset

5.4.2 Ionosphere

This is another dataset taken from the UCI Machine Learning Repository (Frank and Asuncion, 2010). It consists of 351 observations, 24 variables and 2 classes. We will see in the next section that the rule metric and random forest methods perform really well on this dataset.

In this example, we would like to illustrate the lower dimensional representation of the metric. The data is split randomly into testing and training set of size 106 and 245 observations. We first trained our model with the training set. The lower dimensional representation of the learned metric is given in Figure 5.7.

The test accuracy is 91.5% for this particular training/testing set split. The lower dimensional projection shows that some observations of the class "bad" lie in the region where the class "good" dominates. We might want to investigate these observations further.

5.4.3 Performance Comparison

More data sets from the UCI repository are chosen to test the performance of our method.

To compare the effect of learning a distance metric, we consider nearest neighbour using the Euclidean metric on the raw data, neighbourhood component analysis, the method proposed by Xing and our method which we will call rule metric. For simplicity, the number of neighbours is fixed at 3 in the nearest neighbour algorithm. We also compare a few different methods: random forest (Breiman, 2001), linear discriminant analysis, rule ensemble and classification tree. For rule ensemble, we apply the rule ensemble algorithm provided by (Friedman and Popescu, 2008). Their implementation only support binary classification and we hence apply a one-against-all strategy by fitting a classification model for each class. The prediction is given by the label that has the highest predicted probability.

The data are then split randomly into a testing and training set in a 30/70 ratio. The mean test accuracies of 20 runs are reported below

Dataset	Wine	Glass	Ionosphere	Iris	Balance
#observations	178	214	351	150	625
#variables	13	9	24	4	4
#class	3	6	2	3	3
knn (Euclidean raw data)	70.6	64.2	83.8	95.9	84.1
NCA	90.4	64.5	85.6	95.3	94.4
Xing	91.2	51.9	87.4	97.0	88.6
Rule metric	95.7	68.1	92.3	94.0	83.9
CART	84.4	63.2	89.5	93.7	76.1
LDA	98.2	58.8	85.9	98.1	86.7
Random Forest	97.9	75.5	93.2	94.6	85.3
Rule Ensemble	94.1	70.9	92.7	93.6	87.1

Rule metric offers substantial improvements over a single classification tree. In the wine dataset, the improvement is over 10% in test accuracy. Random Forest has a better prediction accuracy, but it is difficult to interpret an ensemble of trees.

Our metric offers improvement over the Euclidean metric based on the original data in the wine, glass and ionosphere dataset. The rule metric also has better predictive accuracy than other metric learning method in these datasets. Rules provide a richer structure to explain interaction than a linear transformation of the data can provide.

The predictive performance of the rule metric is similar to the rule ensemble. Rule metric provides one fitted metric while the rule ensemble provides a set of fitted models. These two methods allows different ways of understanding the data, for example, Rule metric allows one to explore why two observations are different through local influence as demonstrated in section 5.4.1.2 while the set of fitted models provided by rule ensemble allows one to investigate which rules would affect class membership in general.

However, our method does not do very well when the input dimensionality is small, such as the iris and balance data. One reason might be finding a diagonal metric in terms of rules indicator functions effectively maps the data into higher dimensions.

5.5 Summary and Discussion

We formulated the metric learning problem proposed by Xing et al. (2002) as a logistic regression problem. The coordinate descent algorithm proposed by Friedman et al. (2009) was applied to solve our optimization problem. Rule ensembles (Friedman and Popescu, 2008) are used as the basis of metric learning with the distance between observations defined as the rules they differ. Our metric based on rule ensembles has better predictive performance than a single classification tree and remains interpretable. The metric also gives useful lower dimensional representation of the data. The lower dimensional projection and rule importance measure provide an alternative way to explore the data.

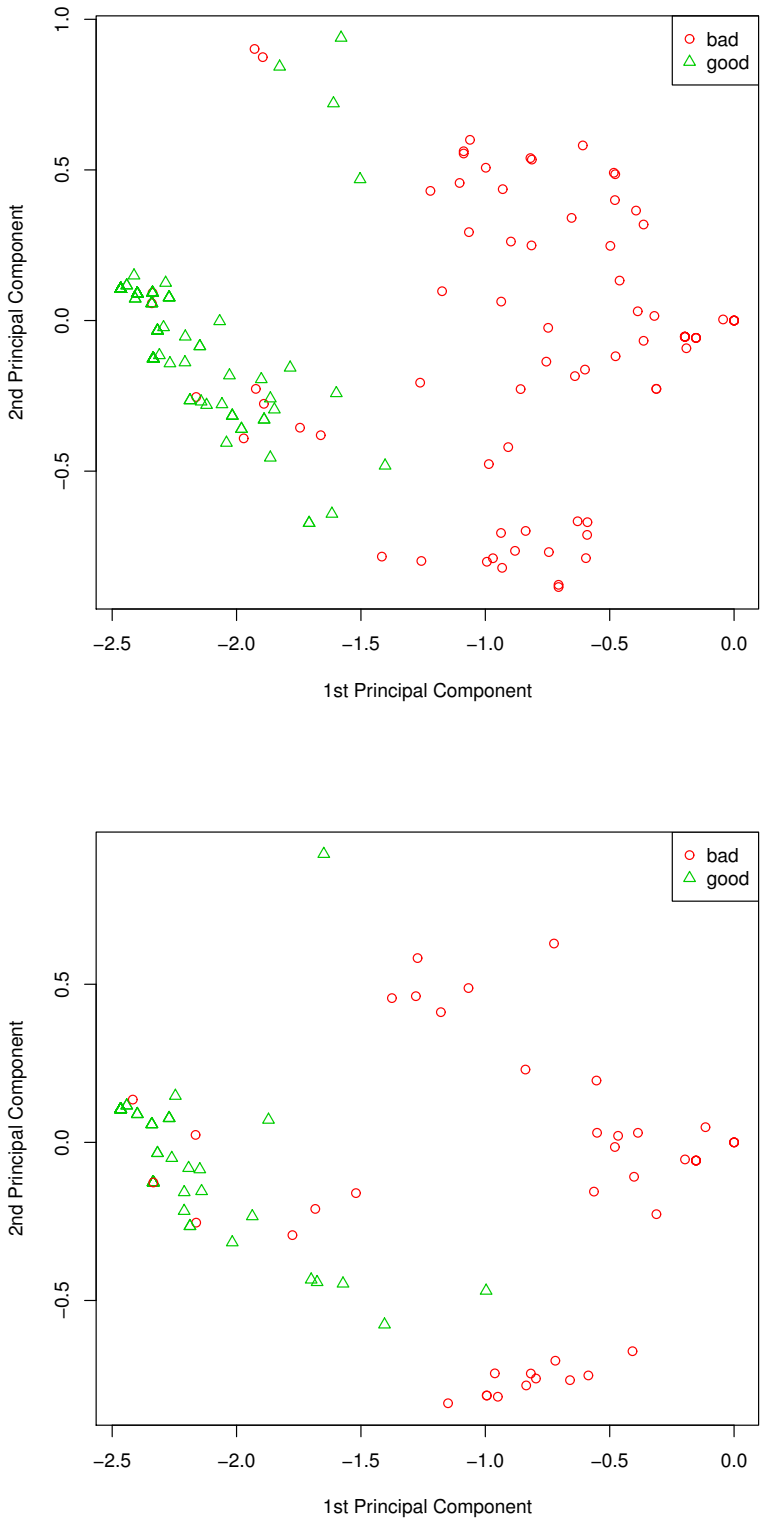


Figure 5.7: The projection of the Ionosphere data given by the learned metric:training set(top), testing set(bottom)

Chapter 6

Application in Document Retrieval

We illustrate an application of distance metric learning in a document retrieval application. Instead of explicit class labels information, the input required for is that the user can decide whether document A is more similar to document B than to document C given three documents A, B and C. Our aim is to learn a metric so that the above relationship is preserved, i.e. $d_{\mathbf{M}}(i,j) < d_{\mathbf{M}}(i,k)$. This motivates a loss function similar to the loss function of Large Margin Nearest Neighbour (Weinberger and Saul, 2009). We demonstrate our algorithm in a multi-label prediction problem.

6.1 Motivation

Given n observations and a $n \times p$ covariate matrix X. Suppose for any triplet of observations (i,j,k), we can tell that observation i is more similar to observation j than to observation k. In other words, we are able to tell whether $d(i,j) < d(i,k)$ is true. Let $\mathcal{T}$ be the set of triplets that

$$d(i,j) < d(i,k) \quad \forall (i,j,k) \in \mathcal{T}.$$

The motivation of our algorithm is to find an Euclidean distance metric so that the ordering is preserved. This can be expressed as the following

$$d_{\mathbf{M}}(i,j) < d_{\mathbf{M}}(i,k) \quad \forall (i,j,k) \in \mathcal{T}$$

where $d_{\mathbf{M}}(i,j) = (X_i - X_j)^T \mathbf{M} (X_i - X_j)$ is the Mahalanobis distances and $\mathbf{M}$ is positive semi-definite.

We add a margin 1 to ensure that the observations are well separated.

$$d_{\mathbf{M}}(i,j) + 1 < d_{\mathbf{M}}(i,k) \quad \forall (i,j,k) \in \mathcal{T}$$

This motivates the minimization of the following loss function.

$$L(\mathbf{M}) = \sum_{(i,j,k)\in\mathcal{T}} [d_{\mathbf{M}}(i,j) + 1 - d_{\mathbf{M}}(i,k)]_+ \tag{6.1}$$

where

$$[z]_+ = \begin{cases} z & \text{if } z \geq 0 \\ 0 & \text{if } z = 0 \end{cases}.$$

This loss function is similar to the loss function proposed in Large Margin Nearest Neighbour (Weinberger and Saul, 2009).

6.2 Similarity as a side-information

The metric learning algorithm required that the user can tell, given three observations, whether observation 1 is more similar to observation 2 than to observation 3. This information can often be easier to obtain than requiring the user to manually label the documents with the appropriate categories. This type of data arises naturally in some recommendation systems. For example, if a user has read articles A and B, this suggests that article A is more similar to article B than other articles. This type of side-information can be applied to learn a distance metric. The fitted distance metric allows for a document retrieval system that returns similar documents. It can also be applied in a semi-supervised classification problem where there are small number of labelled samples.

6.3 Related Work

Our loss function can be seen as an extension of the Large Margin Nearest Neighbour (Weinberger and Saul, 2009) for classification. Given class labels y, the set $\mathcal{T}$ of triplets is defined as

$$\mathcal{T} = \{(i,j,k) \in \mathbb{N}^{n\times n\times n} : y_i = y_j \text{ and } y_i \neq y_j\}.$$

The set $\mathcal{S}$ of similar observations is defined as

$$\mathcal{S} = \{(i,j) \in \mathbb{N}^{n \times n} : y_i = y_j \text{ and } j \text{ is one of the k-nearest neighbours of } i\}.$$

The restriction that j is one of the k-nearest neighbours of i provides some provision for multi-modal class distribution. Weinberger and Saul (2009) proposed to minimize the following loss function

$$L(\mathbf{M}) = \sum_{(i,j)\in\mathcal{S}} d_{\mathbf{M}}(i,j) + \sum_{(i,j,k)\in\mathcal{T}} [d_{\mathbf{M}}(i,j) + 1 - d_{\mathbf{M}}(i,k)]_+ .$$

Chechik et al. (2010) also proposed a similar formulation, but applied this to fit a set of similarities. Let $\mathcal{T}$ be the set of triplets for which observation i is more similar to observation j than to observation k.

$$s(i,k) < s(i,j) \quad \forall (i,j,k) \in \mathcal{T}.$$

where $s(i,k)$ denotes the similarities between observations i and k.

$$s_{\mathbf{W}}(i,k) < s_{\mathbf{W}}(i,j) \quad \forall (i,j,k) \in \mathcal{T}$$

where $s_{\mathbf{W}}(i,j) = X_1 \mathbf{W} X_2$ and $\mathbf{W}$ is a $p \times p$ matrix. The matrix $\mathbf{W}$ is not required to be symmetric or positive semi-definite. They applied their algorithm in an image search engine.

6.4 Stochastic Gradient Descent Algorithm

Since the size of all possible triplets grows at the order $O(n^3)$ where n is the number of observations, an efficient algorithm is required to solve the optimization problem. We found that the stochastic gradient descent algorithm is simple but effective. The stochastic gradient descent algorithm is similar to the ordinary gradient descent algorithm, with the exception that the gradient is computed only from one random triplet in each iteration.

To solve the following optimization problem

$$\min_{\mathbf{M}\succeq 0} \frac{1}{|\mathcal{T}|} \sum_{(i,j,k)\in\mathcal{T}} [d_{\mathbf{M}}(i,j) + 1 - d_{\mathbf{M}}(i,k)]_+ + \Omega(\mathbf{M})$$

where Ω is a suitable regularization penalty, the stochastic gradient descent algorithm states as follows:

1. Initialize $\mathbf{M}_0$.
2. Randomly draw a triplet (i, j, k) from the set $\mathcal{T}$.
3. $\mathbf{M}_{t+1/2} = \mathbf{M}_t - \eta_t \nabla L(i, j, k, \mathbf{M}_t)$, where $\nabla L(i, j, k, \mathbf{M}_t)$ is the gradient computed using the triplet (i, j, k).
4. Project $\mathbf{M}$ back to the positive semi-definite cone with $\mathbf{M}_{t+1} = \Pi(\mathbf{M}_{t+1/2})$.
5. Repeat step 2 to 4 until convergence.

The step size η_t can be chosen to be a small constant. Suppose Ω is the trace norm penalty: $\Omega(\mathbf{M}) = \lambda tr(\mathbf{M})$. Then, the gradient $\nabla L(i, j, k, \mathbf{M}_t)$ can be computed by the following formulae.

If $d_{\mathbf{M}}(i, j) + 1 - d_{\mathbf{M}}(i, k) > 0$,

$$\nabla L(i, j, k, \mathbf{M}_t) = (X_i - X_j)(X_i - X_j)^T - (X_i - X_k)(X_i - X_k)^T + \lambda \mathbf{I}.$$

Otherwise,

$$\nabla L(i, j, k, \mathbf{M}_t) = \lambda \mathbf{I}$$

where $\mathbf{I}$ is a $p \times p$ identity matrix. The projection back to the positive semi-definite cone can be done with an eigendecomposition as described previously in Section 3.6.9. Maintaining positive semi-definiteness is computationally expensive which scales at order $O(p^3)$. For problem in higher dimensions, the restriction to a diagonal metric allows the optimization to be computed within a reasonable time.

The loss function is computed over a small set of training examples to check convergence. Since each iteration can be computed quickly, many iterations can be performed. Although it is very slow to converge in theory, a good solution can often be obtained even when only a fraction of training example has been visited. Zhang (2004) provided some theoretical results on the convergence of stochastic gradient descent algorithm.

6.5 Labelling text data

Reuters Corpus Volume I(RCV1) is a dataset consists of categorized news stories. RCV1-v2 is the processed version of the data given by Lewis et al. (2004). In RCV1-

v2, each document is reduced to a list of stemmed words. A typical representation of a news article would be of the following form:

```
... market life emerg evident evident econom econom ... econom
econom back back back track buzz tuesday tuesday tuesday tuesday
tuesday stock stock ... stock clos record record high high high
interest interest interest interest rate rate ... confound strength
end long long contract drop benchmark auction remain stead feder
reserv refrain ... head research research santand city city continu
declin inflat gdp growth figur lack upward move
```

The news stories were manually tagged as four main topics. Economics, Corporate/Industrial, Government/Social, Markets. Within each topic, it is further divided into subtopics. For example, under the main topics Economics, the subtopics are economic performance, monetary, inflation, consumer finance, government finance, capacity, employment, trade/reserves, housing starts, leading indicators. In the original data, the subtopics are further refined into more sub-categories, but we consider only the first two levels of topics in this discussion. A news article could be labelled with multiple topics and subtopics, for example, the above example article is labelled as Economics - economic performance, Markets - equity markets and Markets - bond markets. There are 55 subtopics in total: 18 under Corporate/Industrial, 10 under Economics, 23 under Government/Social and 4 under Markets. The aim is to predict the labels from the set of tokens and hence this is an example of a multi-label prediction problem.

6.5.1 Hierarchy encoded by dissimilarity

Recall that the set of triplets $\mathcal{T}$ is defined as

$$d(i,j) < d(i,k) \quad \forall (i,j,k) \in \mathcal{T}.$$

The loss function 6.1 depends only on the definition of the set of triplets $\mathcal{T}$. In this section, we define the set of triplets $\mathcal{T}$ for the text data. Since the optimization algorithm described in Section 6.4 requires a random draw from the set $\mathcal{T}$, it would be more constructive to discuss how an triplet is drawn from $\mathcal{T}$:

1. First of all, i is drawn uniformly from $\{1, 2, \ldots, n\}$.
2. j is then drawn uniformly from the set of 50 nearest observations of i(under Euclidean distances) that share the same main categories (Economics, Corporate/Industrial, Government/Social, Markets)

3. k is drawn uniformly from $\{1, 2, \ldots, n\}$ such that $i \neq j \neq k$.

4. If articles i and k do not share the same main categories, then return the triplet (i, j, k).

5. If articles i,j and k all share the same main topics, the relative distances $d(i, j)$ is defined by the hamming distances of the assigned topics.

$$\text{Hamming distance} = \frac{\text{the number of topics that differ}}{\text{the total number of assignable topics}}.$$

   If $d(i, j) < d(i, k)$, then return the triplet (i, j, k). If $d(i, j) > d(i, k)$, return the triplet (i, k, j). If there is a tie $d(i, j) = d(i, k)$, then draw a new triplet.

The restriction that j has to be one of the neighbours of i is influenced by the proposal of large margin nearest neighbour (Weinberger and Saul, 2009). The above scheme effectively defines the scope of $\mathcal{T}$. For example, the following triplet is included in $\mathcal{T}$.

1. Economics
2. Economics
3. Government/Social

while the following triplet is not included as the first two observations do not share the same main topics.

1. Economics
2. Economics and Markets
3. Government/Social

For example, the following triplet is included in $\mathcal{T}$.

1. Economics - monetary, inflation
2. Economics - monetary, inflation
3. Economics - monetary

In the case of a tie in the hamming distances, the corresponding triplet is not included. For example, the following triplet is not included.

1. Economics - monetary
2. Economics - inflation

3. Economics - employment

The above examples demonstrate one advantage of our formulation: it is not necessary to state the dissimilarity explicitly. The only information required is that we can compare dissimilarity between two pairs of observations.

6.5.2 Results

The simulation is performed on the set of news articles published from August 20, 1996 to August 31, 1996. This set of 23149 articles is split into a testing set(20%) and training set (80%). The design matrix X is an indicator matrix of whether a stemmed word exists in a document. The list of tokens is transformed into the an indicator functions whether that token exists in a particular article. Low frequency tokens with less than 50 occurrences are removed. This gives an indicator matrix X with 3769 columns.

A diagonal distance metric $\mathbf{M}$ is trained with the following loss function

$$\frac{1}{|\mathcal{T}|} \sum_{(i,j,k)\in\mathcal{T}} [d_{\mathbf{M}}(i,j) + 1 - d_{\mathbf{M}}(i,k)]_+ + \lambda \|\mathbf{M}\|_1$$

where a suitable regularization parameter is chosen by a held-out validation set taken from the training data.

As a comparison, we train a support vector machine(SVM) for each category separately. Each support vector machine classifier is trained with radial basis function kernel and the parameters are tuned automatically by the R package *kernlab*.

We consider three classifiers: nearest neighbour using the learned metric, nearest neighbour using the raw data and logistic regression trained individually on each topic and subtopic. We compare the quality of the nearest neighbour using Hamming distance between the topics of the nearest neighbour and the test data point.

$$\text{Hamming distance} = \frac{\text{the number of topics that differ}}{\text{the total number of assignable topics}}.$$

	Learned metric	Original	SVM
Average Hamming distance	0.02282	0.02648	0.01747

The Hamming distance equals to the average classification error for each category. Support Vector Machine has the best performance among three classifiers.

Since each article is only tagged with a small number of categories, perhaps a more relevant measure would be the Jaccard distance. We compare the quality of the nearest neighbour using Jaccard distance between the topics of the nearest neighbour and the test data point. The Jaccard distance is defined as

$$\text{Jaccard distance} = 1 - \frac{|\hat{C}_i \cap C_i|}{|\hat{C}_i \cup C_i|}.$$

where C_i is the assigned categories of observation i and $\hat{C}_i$ is the predicted categories of observation i.

	Learned metric	Original	SVM
Average Jaccard Distance	0.2854	0.3426	0.2949

Under this measurement, the metric learning algorithm has similar performance as the support vector machine.

Finally, we compare the accuracy of the classification where the predicted categories exactly match.

	Learned metric	Original	SVM
Accuracy	56.42%	52.75%	50.46%

The nearest neighbour using the learned metric improves on the nearest neighbour based on Euclidean distances in the original space. The learned metric returns a more relevant document on average than using the raw Euclidean metric. The support vector machine does not perform very well under this measure, partly due to the fact that it ignores the hierarchical structure of the classes.

6.6 Scene Recognition

In this section, we consider an application in image classification. This set of data is first analysed by Boutell et al. (2004) for multi-label learning. The dataset consists of 1211 training images and 1196 testing images. Boutell et al. (2004) explained that the luminance-chrominance decomposition removes non-linearity of the standard RGB colour space. Each image is first converted into the CIELUV colour space. Then, each image is divided by a 7×7 grid into 49 blocks. The mean and variance of each colour component within each block are recorded. This gives a feature vector of dimension$= 2 \times 3 \times 49 = 294$. Each image is tagged with at least one of the following categories: Beach, Sunset, Fall Foliage, Field, Mountain, Urban.

6.6.1 Result

We learn a distance metric $\mathbf{M}$ with a trace norm regularization. A triplet is drawn from the set $\mathcal{T}$ with the following algorithm.

1. First of all, i is drawn uniformly from $\{1, 2, \ldots, n\}$.
2. Let S j is then drawn uniformly from the set of 10 nearest observations of i(under Euclidean distances)
3. k is drawn uniformly from $\{1, 2, \ldots, n\}$ such that $i \neq j \neq k$.
4. The relative distances $d(i, j)$ is defined by the hamming distances of the assigned categories. If $d(i, j) < d(i, k)$, then return the triplet (i, j, k). Otherwise, then draw a new triplet.

The performance of nearest neighbour with the test data is given in the following table which shows that metric learning is able to improve the nearest neighbour classifier.

	Learned Metric	Original Distance
Hamming Loss	0.1060	0.1249
Jaccard Loss	0.3134	0.3623
Exact match Accuracy	64.05%	59.11%

In comparison, we first compute a support vector machine classifier for each category and combine the result. We also compute a support vector machine classifier where the we treat the each combination of categories as separate classes: this gives a multi-class classification problem. In theory, there are $2^6 - 1 = 63$ possible combinations, however, only 15 combinations exist in the training data.

	SVM	SVM(Multi-class)
Hamming Loss	0.0833	0.0829
Jaccard Loss	0.3708	0.2446
Exact match Accuracy	59.28%	71.66%

Therefore, the multi-class SVM has the best accuracy among all methods. The metric learning method has an accuracy that lies between the simple SVM model and the multi-class SVM model.

6.7 More on multi-label classification

We demonstrated that the type of similarity information required for the metric learning problem can adapt itself to quite complex scenarios with multiple labels

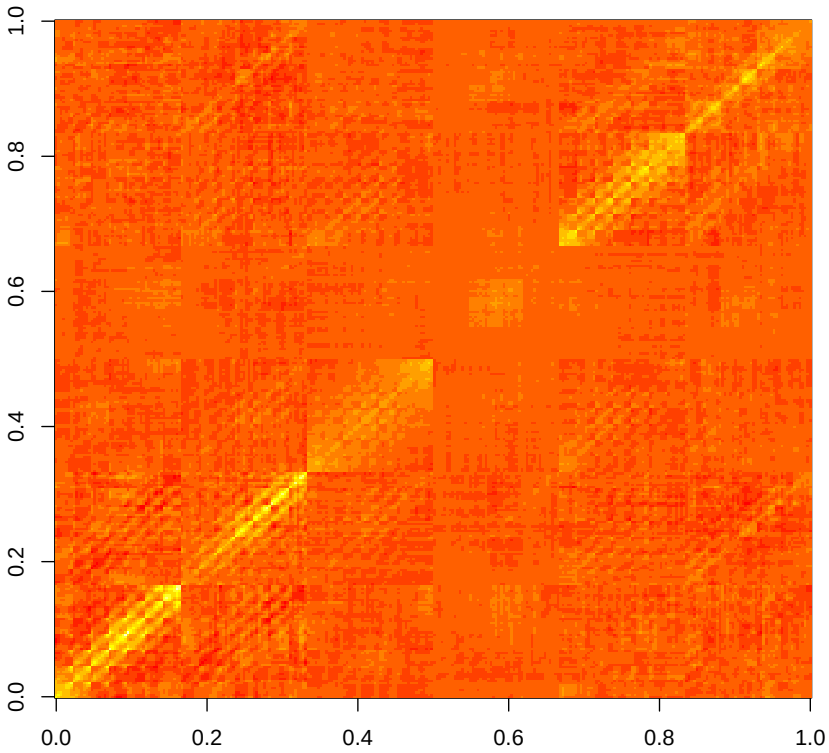


Figure 6.1: The fitted metric $\hat{\mathbf{M}}$ shows an interesting block structure. This has a connection with how the data is generated from a 7×7 grid.

and hierarchical categories. However, we note that both hierarchical classification and multi-label classification are in general quite complex problems. For example, the extension from nearest neighbour to k-nearest neighbour would not be trivial. Zhang and Zhou (2007) suggested one should adjust the cut-off probability for each possible category in a multi-label problem. The predicted labels might violate the hierarchical structure and a more sophisticated scheme would be required to enforce the restriction. Koller and Sahami (1997) proposed to predict the class labels in a top-down fashion that if an observation is not labelled with a category, then it would not be labelled with any of its sub-categories. We refer to the survey paper of Silla and Freitas (2011) and Tsoumakas et al. (2010) for more information on hierarchical classification and multi-label classification.

6.8 Discussion

We applied distance metric learning in a multi-label prediction problem. It can also be applied to learning task where there are no discrete classes or the class labels are unknown, especially in learning task that the main aim is to retrieve similar documents/images rather than classification. This can also be part of a semi-supervised classification scheme when assigning class labels are costly and difficult, while comparing dissimilarity between pairs can be done with relative ease.

Chapter 7

Conclusion

We introduced sparse Mahalanobis distance metric learning and its application in various classification tasks. The aim of Mahalanobis distance metric learning is to learn a positive semi-definite matrix $\mathbf{M}$ such that the accuracy of nearest neighbour classification improves under the pairwise distances defined as $d_{\mathbf{M}}(i,j) = (X_i - X_j)^T \mathbf{M}(X_i - X_j)$. Three different types of sparsity pattern of the matrix $\mathbf{M}$ are investigated: $\mathbf{M}$ consists of many zero entries, $\mathbf{M}$ has many zero rows/columns and $\mathbf{M}$ is low rank and we showed that a lasso penalty $\|\mathbf{M}\|_1$, overlap group lasso penalty $\|\mathbf{M}\|_{group}$ and trace norm penalty $\|\mathbf{M}\|_*$ allows recovery of each type of sparsity pattern respectively. We proved that sparse recovery is possible assuming compatibility condition for each type of penalties. We demonstrated some of the application of distance metric learning in classification. We discovered that learning a diagonal metric $\mathbf{M}$ often gives competitive classification performance in practice and avoids the costly step of maintaining positive semi-definiteness of $\mathbf{M}$ in optimization. The $\ell 1$ regularization allows us to consider an expanded set of basis and we illustrated an application of metric learning with logistic loss and rule ensembles. Rule ensembles allow intuitive interpretation and show good predictive accuracy. We demonstrated how the metric learning problem allows a semi-supervised learning scheme to implement a document retrieval application and multi-label prediction.

Appendix A

Consistency of L1 penalized regression

In this section, we discuss some of the existing theories on the $\ell 1$ penalized regression. For a least squared regression problem with n observations and p variables, let $\mathbf{X}$ be a $n \times p$ design matrix, y be a response vector of length n and β be a coefficient vector of length p. Assume that the responses y are distributed as

$$y = \mathbf{X}\beta^* + \varepsilon$$

where the vector ε is independent and identically distributed noise with mean zero and suppose further that β^* is sparse: β^* has only a few non-zero entries. Let $S_0 = \{i : \beta^* \neq 0\}$ be the set of indices where β^* is non-zero and let $s_0 := |S_0|$ be the cardinality of the set S_0. We consider minimizing the squared error with an $\ell 1$ penalty of the coefficients

$$\hat{\beta} = \text{argmin}_\beta \frac{1}{n}\|y - \mathbf{X}\beta\|_2^2 + \lambda\|\beta\|_1$$

with the aim to show that $\hat{\beta}$ is a good estimate of β^*.

A.1 On the compatibility condition

Successful sparse recovery of β^* would only be possible if the columns of $\mathbf{X}$ are not highly correlated. One of such conditions on the design matrix would be the compatibility condition (Van De Geer and Bühlmann, 2009).

Compatibility Condition The compatibility condition is met for the set S_0, if for some $\phi_0 > 0$, and for all β satisfying $\|\beta_{S_0^c}\|_1 \leq 3\|\beta_{S_0}\|_1$, it holds that

$$\|\beta_{S_0}\|_1^2 \leq (\beta^T \hat{\Sigma} \beta) s_0 / \phi_0^2 \tag{A.1}$$

where

$$\hat{\Sigma} = \mathbf{X}^T \mathbf{X}/n,$$

and β_{S_0} and $\beta_{S_0^c}$ are defined as

$$(\beta_{S_0})_i = \begin{cases} \beta_i & \text{if } i \in S_0 \\ 0 & \text{otherwise} \end{cases},$$
$$\beta_{S_0^c} = \beta - \beta_{S_0}.$$

∎

Another closely related assumption is the restricted eigenvalue condition.

Restricted Eigenvalue The restricted eigenvalue condition is met, if for some $\phi_0 > 0$, and for all β satisfying $\|\beta_{S_0}^c\|_1 \leq 3\|\beta_{S_0}\|_1$, it holds that

$$\|\beta_{S_0}\|_2^2 \leq (\beta^T \hat{\Sigma} \beta)/\phi_0^2. \tag{A.2}$$

∎

The inequality $s_0 \|\beta_{S_0}\|_2^2 \geq \|\beta_{S_0}\|_1^2$ connects these two conditions: if restricted eigenvalue condition holds for some $\phi_0 > 0$, then compatibility condition also holds for the same constant ϕ_0.

If $\hat{\Sigma} = \mathbf{X}^T\mathbf{X}/n$ is invertible, the compatibility constant ϕ^2 can be chosen to be the smallest eigenvalue of $\hat{\Sigma}$. With ϕ^2 being the smallest eigenvalue of $\hat{\Sigma}$, the following holds

$$\frac{\beta^T \hat{\Sigma} \beta}{\|\beta\|_2^2} \geq \phi^2$$

and therefore restricted eigenvalue condition holds and hence the compatibility conditon holds. If $\hat{\Sigma}$ is not invertible, then a positive compatibility constant is still possible since compatibility condition only needs to hold over a cone $\|\beta_{S_0^c}\|_1 \leq 3\|\beta_{S_0}\|_1$. The connection between compatibility condition with other conditions are discussed in detail in Van De Geer and Bühlmann (2009).

One rather interesting theorem proved in Van De Geer and Bühlmann (2009) states that if compatibility condition holds for a covariance matrix Σ_0, then compatibility

condition also holds for another covariance matrix Σ_1 if the difference between them is small.

Theorem A.1.1 *Suppose that the Σ_0 compatibility condition holds for the set S with cardinality s, with compatibility constant $\phi^2_{\Sigma_0}(S)$, and that $\|\Sigma_1 - \Sigma_0\|_\infty \leq \lambda$, where*

$$32\lambda s/\phi^2_{\Sigma_0}(S) \leq 1.$$

Then, for the set S, the Σ_1-compatibility condition holds as well, with $\phi^2_{\Sigma_1}(S) \geq \phi^2_{\Sigma_0}(S)/2$.

$\|.\|_\infty$ denotes the infinity norm:

$$\|\Sigma_1 - \Sigma_0\|_\infty = \max_{ij} |(\Sigma_1)_{ij} - (\Sigma_0)_{ij}|.$$

This result allows us to derive that compatibility condition holds for a wide range of random data matrix $\mathbf{X}$. Suppose we have a design matrix X where each entry is generated by a bounded, independent and identical distribution with mean 0 and variance 1. The expected covariance matrix would then be $\Sigma_0 := \mathbb{E}(\mathbf{X}^T\mathbf{X}/n) = \mathbf{I}$ the identity matrix. The compatibility constant $\phi^2_{\Sigma_0}$ of Σ_0 is lower bounded by the smallest eigenvalue of Σ_0 which is 1. Let Σ_1 be the empirical covariance matrix $\mathbf{X}^T\mathbf{X}/n$. The empirical covariance matrix is tightly concentrated around the expected covariance matrix by Hoeffding's inequality as the entries of X are bounded. Hence, one could show that $\|\Sigma_1 - \Sigma_0\|_\infty \leq \lambda$ holds with high probability with the choice of $\lambda = O(\sqrt{log(p)/n})$. With this choice of λ and theorem A.1.1, one could derive that the compatibility condition holds with high probability for a random matrix when $n \gg p$.

A.2 Consistency Theorem

The consistency theorem in Van De Geer and Bühlmann (2009) states as follows. Let

$$\mathbb{J} := \left\{ \max_{1 \leq j \leq p} 2|\varepsilon^T X^{(j)}|/n \leq \lambda_0 \right\}$$

where $X^{(j)}$ denotes the j-th column of $\mathbf{X}$.

Theorem A.2.1 (Consistency of lasso in Van De Geer and Bühlmann (2009)) *Suppose the compatibility condition holds. Then on $\mathbb{J}$, we have for $\lambda \geq 2\lambda_0$,*

$$\|X(\hat{\beta} - \beta^*)\|_2^2/n + \lambda\|\hat{\beta} - \beta^*\|_1 \leq 4\lambda^2 s_0/\phi_0^2.$$

Under suitable condition, λ_0 can be chosen at $O(\sqrt{\log p/n})$ such that $\mathbb{J}$ holds with high probability. For example, if we assume that the covariates are bounded by c $|\mathbf{X}_{i,j}| \leq c$ and the noise has mean zero and bounded by δ, $|\varepsilon_i| \leq \delta$, then by Hoeffding's inequality, for any $j = 1, \ldots, p$,

$$P(2|\varepsilon^T X^{(j)}|/n \geq \lambda_0) \leq 2 \exp\left(-\frac{\lambda_0^2 n}{16c^2\delta^2}\right).$$

and applying a union bound on the probabilities gives

$$P\left(\max_{1\leq j\leq p} 2|\varepsilon^T X^{(j)}|/n \leq \lambda_0\right) \leq 1 - 2p \exp\left(-\frac{\lambda_0^2 n}{16c^2\delta^2}\right).$$

Therefore, λ_0 can be chosen of order $O(\sqrt{\log p/n})$. Combined with the choice of $\lambda_0 = O(\sqrt{\log p/n})$, the consistency theorem shows that the mean squared error converges at the rate of $O(\log p/n)$ and the $\ell 1$ distance between $\hat{\beta}$ and β^* converges at the rate of $O(\sqrt{\log p/n})$.

A.3 Sign consistency

The consistency theorem proved in this thesis would be in the form of theorem A.2.1. Here, we briefly discuss another type of consistency called sign consistency and how one could derive sign consistency from theorem A.2.1.

Sign consistency is another type of consistency result which aims to prove that

$$\text{sign}(\hat{\beta}_i) = \text{sign}(\beta_i^*)$$

where

$$\text{sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}.$$

Zhao and Yu (2006) proved that such recovery is possible with $\ell 1$ penalty only if irrepresentable condition holds. However, irrepresentable condition is a more restrictive condition than compatibility condition as Van De Geer and Bühlmann (2009) showed that irrepresentable condition implies compatibility condition.

Although theorem A.2.1 does not provide sign consistency directly, it shows that the $\ell 1$ distance between $\hat{\beta}$ and β^* $\|\hat{\beta} - \beta^*\|_1$ is small. This property can be exploited to achieve sign consistency. For example, Meinshausen and Yu (2009) suggested

removing fitted coefficients of small magnitude, and Zou (2006) proposed adaptive lasso where the lasso regularization parameters for each variable is adjusted according to the magnitude of the fitted coefficients of the initial fit.

Bibliography

Bach, F. R. (2008). Consistency of trace norm minimization. *The Journal of Machine Learning Research 9*, 1019–1048.

Bertsekas, D. P. (1999). *Nonlinear Programming* (2nd ed.). Athena Scientific.

Bian, W. and D. Tao (2011). Learning a distance metric by empirical loss minimization. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Two*, pp. 1186–1191. AAAI Press.

Bickel, P., Y. Ritov, and A. Tsybakov (2009). Simultaneous analysis of Lasso and Dantzig selector. *Annals of Statistics 37*, 1705–1732.

Boutell, M. R., J. Luo, X. Shen, and C. M. Brown (2004). Learning multi-label scene classification. *Pattern recognition 37*(9), 1757–1771.

Boyd, S., N. Parikh, E. Chu, B. Peleato, and J. Eckstein (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning 3*(1), 1–122.

Boyd, S. and L. Vandenberghe (2004). *Convex Optimization*. Cambridge University Press.

Breiman, L. (2001). Random Forests. *Machine Learning 45*, 5–32.

Bühlmann, P. and S. van de Geer (2011). *Statistics for High-dimensional Data*. Springer.

Candès, E. J. and Y. Plan (2010). Matrix completion with noise. *Proceedings of the IEEE 98*(6), 925–936.

Candès, E. J. and B. Recht (2009). Exact matrix completion via convex optimization. *Foundations of Computational mathematics 9*(6), 717–772.

Candès, E. J. and T. Tao (2005). Decoding by linear programming. *Information Theory, IEEE Transactions on 51*(12), 4203–4215.

Chechik, G., V. Sharma, U. Shalit, and S. Bengio (2010). Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research 11*, 1109–1135.

Chesneau, C. and M. Hebiri (2008). Some theoretical results on the grouped variables lasso. *Mathematical Methods of Statistics 17*(4), 317–326.

Dasgupta, S. and A. Gupta (2003). An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Struct. Algorithms 22*(1), 60–65.

Datar, M. and P. Indyk (2004). Locality-sensitive hashing scheme based on p-stable distributions. In *In SCG '04: Proceedings of the twentieth annual symposium on Computational geometry*, pp. 253–262. ACM Press.

Efron, B., T. Hastie, L. Johnstone, and R. Tibshirani (2004). Least angle regression. *Annals of Statistics 32*, 407–499.

Frank, A. and A. Asuncion (2010). UCI machine learning repository.

Friedman, J., T. Hastie, H. Höfling, and R. Tibshirani (2007). Pathwise coordinate optimization. *Annals of Applied Statistics 2*, 302–332.

Friedman, J., T. Hastie, and R. Tibshirani (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics 9*(3), 432–441.

Friedman, J., T. Hastie, and R. Tibshirani (2009). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software 33*(1), 1–22.

Friedman, J. and B. E. Popescu (2008). Predictive learning via rule ensembles. *The Annals of Applied Statistics 2*(3), 916–954.

Fu, W. J. (1998). Penalized regressions: The bridge versus the lasso. *Journal of Computational and Graphical Statistics 7*(3), 397–416.

Goldberger, J., S. Roweis, G. Hinton, and R. Salakhutdinov (2008). Neighborhood component analysis. In *NIPS*.

Goldstein, T. and S. Osher (2009). The split bregman method for l1-regularized problems. *SIAM J. Img. Sci. 2*(2), 323–343.

Golub, G. H. and C. F. Van Loan (2012). *Matrix computations*, Volume 3. JHU Press.

Hastie, T. and R. Tibshirani (1996). Discriminant adaptive nearest neighbor classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 18*(6), 607–616.

Hastie, T., R. Tibshirani, and J. H. Friedman (2003). *The Elements of Statistical Learning*. Springer.

Higham, N. J. (1988). Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra and its Applications 103*, 103 – 118.

Hix, S., A. Noury, and G. Roland (2006). Dimensions of politics in the European Parliament. *American Journal of Political Science 50*, 494–511.

Hoefling, H. (2010). A path algorithm for the fused lasso signal approximator. *Journal of Computational and Graphical Statistics 19*(4), 984–1006.

Hoerl, A. E. and R. W. Kennard (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics 12*(1), 55–67.

Huang, J. and T. Zhang (2010). The benefit of group sparsity. *The Annals of Statistics 38*(4), 1978–2004.

Huang, K., Y. Ying, and C. Campbell (2009). GSML: A unified framework for sparse metric learning. *Data Mining, IEEE International Conference on 0*, 189–198.

Indyk, P. and R. Motwani (1998). Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 604–613. ACM.

Jacob, L., G. Obozinski, and J.-P. Vert (2009). Group lasso with overlap and graph lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 433–440. ACM.

Johnson, W. B. and J. Lindenstrauss (1984). Extensions of Lipschitz mapping into Hilbert space. In *Conf. in modern analysis and probability*, Volume 26 of *Contemporary Mathematics*, pp. 189–206. American Mathematical Society.

Koller, D. and M. Sahami (1997). Hierarchically classifying documents using very few words. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, San Francisco, CA, USA, pp. 170–178. Morgan Kaufmann Publishers Inc.

Krahmer, F. and R. Ward (2011). New and improved Johnson-Lindenstrauss embeddings via the Restricted Isometry Property. *SIAM Journal on Mathematical Analysis 43*, 1269–1281.

Kuhn, H. W. and A. W. Tucker (1951). Nonlinear programming. In *Proceedings of the second Berkeley symposium on mathematical statistics and probability*, Volume 5. California.

Lewis, D. D., Y. Yang, T. G. Rose, and F. Li (2004). Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research 5*, 361–397.

McCullagh, P. and J. A. Nelder (1989). *Generalized linear models (Second edition)*. London: Chapman & Hall.

Meier, L., S. Van De Geer, and P. Buhlmann (2008). The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70*(1), 53–71.

Meinshausen and B. Yu (2009). Lasso-type recovery of sparse representations from highdimensional data. *Annals of Statistics*, 246–270.

Mirsky, L. (1975). A trace inequality of John von Neumann. *Monatshefte für Mathematik 79*(4), 303–306.

Negahban, S. and M. J. Wainwright (2011). Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *Annals of Statistics 39*, 1069–1097.

Osborne, M., B. Presnell, and B. Turlach (2000). A new approach to variable selection in least squares problems. *IMA journal of numerical analysis 20*(3), 389.

Ripley, B. D. (1996). *Pattern recognition and neural networks*. Cambridge: Cambridge University Press.

Rosales, R. and G. Fung (2006). Learning sparse metrics via linear programming. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 367–373. ACM.

Roweis, S. T. and L. K. Saul (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science 290*, 2323–2326.

Saha, A. and A. Tewari (2010). On the finite time convergence of cyclic coordinate descent methods. *arXiv preprint*.

Shalev-Shwartz, S. and A. Tewari (2009). Stochastic methods for l1 regularized loss minimization. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, New York, NY, USA, pp. 929–936. ACM.

Silla, Jr., C. N. and A. A. Freitas (2011). A survey of hierarchical classification across different application domains. *Data Min. Knowl. Discov. 22*(1-2), 31–72.

Slater, M. (1959). Lagrange multipliers revisited. Cowles Foundation Discussion Papers 80, Cowles Foundation for Research in Economics, Yale University.

Soifer, A., B. Grünbaum, P. Johnson, and C. Rousseau (2008). *The Mathematical Coloring Book: Mathematics of Coloring and the Colorful Life of its Creators*. Springer.

Tenenbaum, J. B., V. Silva, and J. C. Langford (2000). A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 2319–2323.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B 58*, 267–288.

Tibshirani, R., M. Saunders, S. Rosset, J. Zhu, and K. Knight (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67*(1), 91–108.

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications 109*, 475–494.

Tsoumakas, G., I. Katakis, and I. Vlahavas (2010). Mining multi-label data. In *Data mining and knowledge discovery handbook*, pp. 667–685. Springer.

Van De Geer, S. A. and P. Bühlmann (2009). On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics 3*, 1360–1392.

Venables, W. N. and B. D. Ripley (2002). *Modern applied statistics with S-Plus*. Springer-Verlag, New York.

Weinberger, K. Q. and L. K. Saul (2009). Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research 10*, 207–244.

Wu, T. T. and K. Lange (2008). Coordinate descent algorithms for lasso penalized regression. *Ann. Appl. Stat. 2*(1), 224–244.

Xing, E. P., A. Y. Ng, M. I. Jordan, and S. Russell (2002). Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, pp. 505–512. MIT Press.

Yuan, M. and Y. Lin (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68*(1), 49–67.

Zhang, M.-L. and Z.-H. Zhou (2007). Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition 40*(7), 2038–2048.

Zhang, T. (2004). Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 116. ACM.

Zhang, T. (2011). Adaptive forward-backward greedy algorithm for learning sparse representations. *Information Theory, IEEE Transactions on 57*(7), 4689–4708.

Zhao, P., G. Rocha, and B. Yu (2009). The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics 37*(6A), pp. 3468–3497.

Zhao, P. and B. Yu (2006). On model selection consistency of lasso. *Journal of Machine Learning Research 7*, 2541–2563.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association 101*(476), 1418–1429.

Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67*(2), 301–320.