ABSTRACT

Title of dissertation: REGULARIZATION METHODS FOR HIGH-DIMENSIONAL INFERENCE

David A. Shaw, Doctor of Philosophy, 2014

Dissertation directed by: Professor Ramalingam Chellappa
Department of Electrical Engineering

High dimensionality is a common problem in statistical inference, and is becoming more prevalent in modern data analysis settings. While often data of interest may have a large – often unmanageable – dimension, modifications to various well-known techniques can be made to improve performance and aid interpretation. We typically assume that although predictors lie in a high-dimensional ambient space, they have a lower-dimensional structure that can be exploited through either prior knowledge or estimation.

In performing regression, the structure in the predictors can be taken into account implicitly through regularization. In the case where the underlying structure in the predictors is known, using knowledge of this structure can yield improvements in prediction. We approach this problem through regularization using a known projection based on knowledge of the structure of the Grassmannian. Using this projection, we can obtain improvements over many classical and recent techniques in both regression and classification problems with only minor modification to a typical least squares problem.

The structure of the predictors can also be taken into account explicitly through methods of dimension reduction. We often wish to have a lower-dimensional representation of our data in order to build potentially more interpretable models or to explore possible connections between predictors. In many problems, we are faced with data that does not have a similar distribution between estimating the model parameters and performing prediction. This results in problems when estimating a lower-dimensional structure of the predictors, as it may change. We pose methods for estimating a linear dimension reduction that will take into account these discrepancies between data distributions, while also incorporating as much of the information as possible in the data into construction of the predictor structure. These methods are built on regularized maximum likelihood and yield improvements in many cases of regression and classification, including those cases in which predictor dimension changes between training and testing.

REGULARIZATION METHODS FOR HIGH-DIMENSIONAL INFERENCE

by

David Andrew Shaw

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2014

Advisory Committee:
Professor Ramalingam Chellappa, Chair/Advisor
Professor Paul J. Smith
Professor Wojciech Czaja
Professor Yuan Liao
Professor John J. Benedetto, Dean's Representative

Acknowledgments

I am grateful for the support and insight of my advisor, Rama Chellappa, and the partial support of the Multi-University Research Initiative (MURI) from the Office of Naval Research under the grant 1141221258513. Additional thanks go to Paul Smith for his many helpful comments on the present document, resulting in a number of improvements.

I extend further thanks to those who helped me along in my work, either through discussing pathologies over coffee, offering interesting puzzles over lunch, or encouraging me to spend my free time doing anything but research. Lastly, I thank Sir Ronald Fisher, John W. Tukey, and C. R. Rao – as well as their contemporaries and successors – for endless sources of inspiration. Research has proved to be an exhilarating process, and I hope that my work can help to uncover even the roughest gem of innovation. As Heraclitus said in *Fragment 10* [1]: "Nature loves to hide."

Table of Contents

List of Tables

List of Figures

1

Introduction

1.1 Motivation

1.1.1 High-Dimensional Inference

We concern ourselves with the problem of statistical inference where we assume to have *predictors* $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$, each independent and distributed as some random variable $\mathbf{X}$. Corresponding to each predictor we assume to also have *responses* $y_1, \ldots, y_n \in \mathcal{Y}$ distributed as some random variable Y. Depending on the problem, $\mathcal{Y}$ will be taken to be either $\mathbb{R}$ or some set $\{1, \ldots, C\}$ for C a fixed, finite constant.

A wealth of problems in statistics and data analysis succumb to problems related to the "curse of dimensionality" [5]. In statistics, one view of this issue of dimensionality is that if p is large and we wish to estimate the *regression function*

$$m(\mathbf{x}) = E[Y \,|\, \mathbf{X} = \mathbf{x}]$$

given our data – where $E[\cdot]$ denotes expectation – we run into problems of stability for small samples. Moreover, as the number of samples increases to infinity, the estimate for the function m will converge slowly to the true m [6].

Another issue seen with high-dimensional data is in the interpretation of

given data through visualization. In [7], we see that if the predictors occupy a p-dimensional hypersphere and are distributed uniformly, the majority of points would be situated near the hypersurface at the "edge" of the hypersphere. However, constructing a two-dimensional projection of these p-dimensional observations would result in a circular cross-section with a high density of points in the center. This could potentially lead to misleading interpretations, and alternative projections may be desired. We will seek methods to ameliorate these effects that a high dimensionality causes in data analysis problems.

1.1.2 Lower-Dimensional Structure

Often in high-dimensional data analysis problems, we assume a structured dependency among the predictors. A way to formalize this structure is to assume high-dimensional data points lie on a *manifold* of dimension d where $d \leq p$ [8]. Manifolds can be either unspecified and estimated from the data, or specified by construction of the predictors.

Formally, we define a manifold as a metric space $\mathcal{X}$ with the property that if $\mathbf{x} \in \mathcal{X}$ then there is some neighborhood $\mathbf{U}$ of $\mathbf{x}$ and some integer $d \geq 0$ so that a homeomorphism exists between $\mathbf{U}$ and $\mathbb{R}^d$. Here, the term *homeomorphism* denotes a bijection between two metric spaces that is continuous and has a continuous inverse.

Often, a manifold is assumed to have a local *chart* $\phi : \mathcal{B}^d_{0,r} \to (\mathcal{B}^p_{x_0,R} \cap \mathcal{X})$ where $\mathcal{B}^d_{0,r}$ is the d-dimensional Euclidean epsilon ball about 0 with radius r, we take $r, R > 0$ as "small," and ϕ is continuously differentiable and bijective. With

this definition of a chart, we can define the manifold $\mathcal{X}$ at a specific point $\mathbf{x}_0$ as

$$\mathcal{X}_{\mathbf{x},0} = \{\phi(\mathbf{u}) \in \mathcal{B}^p_{x_0,R} \subset \mathbb{R}^p : \mathbf{u} \in \mathcal{B}^d_{0,r} \subset \mathbb{R}^d\}.$$

This definition of a manifold is typically helpful for performing statistics on unknown manifolds [9].

One manifold that can be obtained through a special construction of the predictors is the *Grassmannian* $\mathcal{G}(r, s)$. The space $\mathcal{G}(r, s)$ is thought of as the space of all s-dimensional subspaces of $\mathbb{R}^r$. As an example, if we have a full rank matrix $\mathbf{A} \in \mathbb{R}^{r\times s}$, then span($\mathbf{A}$) is an element of $\mathcal{G}(r, s)$. Formally, we take the Grassmannian as the quotient space

$$\mathcal{G}(r, s) = \mathcal{R}(r, s)/\sim$$

where $\mathcal{R}(r, s)$ is the space of all $r \times s$ matrices of rank s, and, for $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{r\times s}$, $\mathbf{U} \sim \mathbf{V}$ if there exists a nonsingular $\mathbf{L} \in \mathbb{R}^{s\times s}$ such that $\mathbf{V} = \mathbf{U}\,\mathbf{L}$ [2].

We will outline two useful constructions on the Grassmannian: the geodesic flow, and the exponential map. For two points $\mathbf{Q}$ and $\mathbf{R}$ on $\mathcal{G}(r, s)$, we have $\mathbf{Q}, \mathbf{R} \in \mathbb{R}^{r\times s}$ and their orthogonal complements as $\mathbf{Q}^\perp, \mathbf{R}^\perp \in \mathbb{R}^{r\times(r-s)}$. Then we write the *geodesic flow* as $\boldsymbol{\delta} : [0, 1] \to \mathcal{G}(r, s)$ with

$$\boldsymbol{\delta}(t; \mathbf{Q}, \mathbf{R}) = \mathbf{Q}\,\mathbf{U}_1\,\boldsymbol{\Gamma}(t) - \mathbf{Q}^\perp\,\mathbf{U}_2\,\boldsymbol{\Sigma}(t) \tag{1.1}$$

where $\mathbf{U}_1, \mathbf{U}_2, \boldsymbol{\Gamma}, \boldsymbol{\Sigma}$ are given by the generalized singular value decomposition

$$\mathbf{Q}^T \mathbf{R} = \mathbf{U}_1 \mathbf{\Gamma} \mathbf{V}^T, \ (\mathbf{Q}^\perp)^T \mathbf{R} = -\mathbf{U}_2 \mathbf{\Sigma} \mathbf{V}^T . \tag{1.2}$$

We define $\mathbf{\Gamma}(t)$ and $\mathbf{\Sigma}(t)$ as diagonal matrices with $\cos(t\theta_i)$ and $\sin(t\theta_i)$ on the diagonal for $i = 1, \ldots, s$ and $\mathbf{\Gamma}$ and $\mathbf{\Sigma}$ are $\mathbf{\Gamma}(1)$ and $\mathbf{\Sigma}(1)$, respectively.

The exponential map and inverse exponential map are functions

$$\exp(\cdot; \boldsymbol{\mu}) : \mathbb{R}^{s(r-s)} \to \mathcal{G}(r, s), \ \exp^{-1}(\cdot; \boldsymbol{\mu}) : \mathcal{G}(r, s) \to \mathbb{R}^{s(r-s)},$$

defined at a point $\boldsymbol{\mu}$ on the Grassmannian where $\mathbb{R}^{s(r-s)}$ is the tangent space to $\mathcal{G}(r, s)$ at the point $\boldsymbol{\mu}$. These functions are useful for mapping between the Grassmannian and the tangent space about a point on the Grassmannian. Both the exponential map and the inverse exponential map can be computed using the geodesic flow above in a computationally efficient manner [10].

1.1.3 Dimension Reduction

The manifold assumption can simplify matters on the theoretical level, but there are still two issues. First, finding an embedding is not necessarily a trivial task. Second, once an embedding is found, using these lower-dimensional points to build models that can be accurately interpreted in the higher-dimensional ambient space is not always possible with these projection methods; that is, if points are explicitly embedded into $\mathcal{X}$, some information that may be useful in the regression may be lost.

We focus here on the first issue, while the second issue will be discussed in

more detail in Chapters 2 and 3. A common remedy to problems in which predictors have a high dimension and are assumed to have some lower-dimensional – typically unknown – structure is to seek a transformation of these predictors into some lower-dimensional space. Since often the predictors will have a structured dependency, the hope is that we will be able to obtain a reduction of the data that does not discard information that we want, or discards as little of this information as possible. We assume to have independent observations $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$ distributed as the random variable $\mathbf{X}$ and define the data matrix

$$\mathbb{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}.$$

In some cases we will be interested in mean-centered data, though we do not wish to assume data has zero mean in general.

Principal component analysis (PCA, [11]) is a classical dimension reduction tool in which we seek a linear dimension reduction parameter $\boldsymbol{\eta} \in \mathbb{R}^{p \times d}$, such that $\boldsymbol{\eta}$ projects the predictors into directions of maximum variation.

$$\hat{\boldsymbol{\eta}} = \arg\max_{\boldsymbol{\eta}} \ \mathrm{tr}\{\boldsymbol{\eta}^T \, \boldsymbol{\Sigma}^{\mathrm{x}} \, \boldsymbol{\eta}\}, \ \text{such that} \ \ \boldsymbol{\eta}^T \, \boldsymbol{\eta} = \mathbf{I}_{\mathrm{d}}$$

where $\boldsymbol{\Sigma}^{\mathrm{x}} = \mathbb{X}^T \, \mathbb{X}$ is proportional to the covariance matrix of the data $\mathbb{X}$, $\mathbf{I}_{\mathrm{d}}$ is the d-dimensional identity matrix, and $\mathrm{tr}\{\cdot\}$ denotes matrix trace. Proceeding with the optimization above by incorporating the constraint through Lagrange multipliers,

the solution $\hat{\boldsymbol{\eta}}$ has the property that

$$\boldsymbol{\Sigma}^{\mathrm{x}}\,\hat{\boldsymbol{\eta}} = \lambda\hat{\boldsymbol{\eta}}$$

where $\lambda > 0$ is a constant. In order to project into the directions of maximum variation of $\mathbb{X}$, we take $\hat{\boldsymbol{\eta}}$ as the d eigenvectors corresponding to the largest d eigenvalues of $\boldsymbol{\Sigma}^{\mathrm{x}}$.

It will be helpful to consider PCA as a maximum likelihood estimator with respect to some likelihood function. Assume $\mathbf{x}_1 \ldots, \mathbf{x}_n \sim \mathbf{X}$ are independent and $\mathbf{X}$ has a corresponding parameterized density function $f(\mathbf{X}; \boldsymbol{\beta})$ for some unknown parameter $\boldsymbol{\beta}$; then the *likelihood function* of the data $\mathbb{X}$ is defined as

$$L(\boldsymbol{\beta}; \mathbb{X}) = \prod_{i=1}^{n} f(\mathbf{x}_{\mathrm{i}}; \boldsymbol{\beta})$$

where interest will lie in estimating the parameter $\boldsymbol{\beta}$. To use this approach for PCA, we first pose an error model for a single observation $\mathbf{X}$ as

$$\mathbf{X} = \boldsymbol{\mu}^{\mathrm{x}} + \boldsymbol{\eta}\,\boldsymbol{\nu} + \boldsymbol{\epsilon},\ \boldsymbol{\epsilon} \sim N(\mathbf{0}, \boldsymbol{\Delta})$$

where $\boldsymbol{\epsilon}$ is a vector of random errors, $\boldsymbol{\nu}$ are unknown coefficients – here equal to $\boldsymbol{\eta}^T(\mathbf{X} - \boldsymbol{\mu}^{\mathrm{x}})$ – and $\boldsymbol{\Delta} > 0$ is a covariance matrix. Setting $\boldsymbol{\Delta} = \sigma^2\,\mathbf{I}_{\mathrm{p}}$, we see that the log-likelihood function for $\boldsymbol{\eta}$ is

$$L(\boldsymbol{\eta}; \mathbb{X}) = -\frac{np}{2}\log(2\pi) - \frac{np}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}[\tilde{\mathbb{X}}^T(\mathbf{I}_{\mathrm{p}} - \boldsymbol{\eta}\,\boldsymbol{\eta}^T)\tilde{\mathbb{X}}].$$

where $\tilde{\mathbb{X}}$ is the mean-centered version of $\mathbb{X}$. Suppressing constant terms and those not involving $\boldsymbol{\eta}$, we see that

$$\hat{\boldsymbol{\eta}} = \arg\max_{\boldsymbol{\eta}} \; L(\boldsymbol{\eta}), \text{ such that } \boldsymbol{\eta}^T \boldsymbol{\eta} = \mathbf{I}_{\mathrm{d}}$$

is equivalent to

$$\hat{\boldsymbol{\eta}} = \arg\max_{\boldsymbol{\eta}} \; \mathrm{tr}\{\boldsymbol{\eta}\, \tilde{\boldsymbol{\Sigma}}^{\mathrm{x}}\, \boldsymbol{\eta}\}, \text{ such that } \boldsymbol{\eta}^T \boldsymbol{\eta} = \mathbf{I}_{\mathrm{d}}$$

where $\tilde{\boldsymbol{\Sigma}}^{\mathrm{x}}$ is the mean-centered covariance matrix. The estimate $\hat{\boldsymbol{\eta}}$ here coincides with the PCA estimate above.

We are typically interested in the subspace span($\boldsymbol{\eta}$), with $\boldsymbol{\eta}$ simply a specific basis for this subspace. In this case, $\boldsymbol{\eta}$ can be thought of as an element of the Grassmannian $\mathcal{G}(p, d)$. This fact is often used to construct various optimization problems for dimension reduction [12].

An early example of PCA applied to a regression problem is given in [13] in which the physical properties of pit props – lengths of lumber used to buttress walls in a mine – are estimated with numerous predictors that are highly correlated. PCA is used to investigate the effect a new set of uncorrelated predictors has on the regression.

PCA is a useful method for obtaining the dimension reduction $\boldsymbol{\eta}$, though as seen above it can be seen to make a number of assumptions when posing it as a maximum likelihood estimate, namely that of isotropic errors. A large body of work exists to develop methods to perform dimension reduction in slightly more

sophisticated ways by learning an approximation to the manifold that the data of interest supposedly lie on. Many popular techniques (e.g., LLE [14], ISOMAP [15]) work by localizing the data in an attempt to take advantage of the local Euclidean structure of the manifold.

1.1.4 Inference in Computer Vision

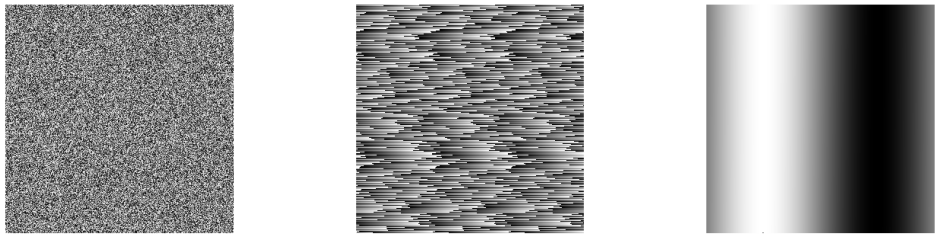
Consider the problem of pattern recognition in which the space of data consists of images. A naïve approach would require modeling the images by treating the grayscale level of each pixel as a separate predictor. In other words, if we scale the values to lie in the unit interval, for one observation $\mathbf{x}$ we have $\mathbf{x} \in [0,1]^p$ where p is the number of pixels in the image. Making accurate predictions will require the image to have a suitably dense resolution, but this quickly becomes a problem as it results in a large number of pixels yielding an unmanageably large number of predictors. More accuracy could be obtained using color images, but this would introduce still more dimensions.

The manifold assumption arises naturally in computer vision. Purely data-dependent methods such as estimating a face subspace using PCA on the difference between each data point and a test image yields promising results [16], though these dimension reduction techniques are mainly concerned with linear embeddings as in PCA above. Incorporating prior knowledge of an image's structure can be done as well. For example, in a "cartoon" image – that is, an black and white image with only smoothly varying borders between black and white portions – for

a sufficiently localized patch the image can be parameterized with the distance of a straight edge from the center of the patch and the angle the straight edge makes with the horizontal axis [17]. More complicated manifolds, such as those that accurately describe images with rich texture, can be modeled with different parameterizations.

A common problem in computer vision concerns using a dataset of images to estimate a high-dimensional regression function. The ISOMAP face dataset [15] consists of 698 images of size 64×64 of a synthetic face at various pose angles and illuminations meaning the naïve approach would result in a set of predictors in $\mathbb{R}^{4096}$, obtained by concatenating the rows of each image into a large vector.

The above results in a problem in which the dimension of the predictors becomes unwieldy. Though this can be a serious issue, many problems will have predictors that are highly dependent upon one another allowing for the possibility of ameliorating the effects this high dimensionality causes by making adjustments to the model. Unless an image contains only white noise, each pixel will depend on every other pixel in the image in some way, with some of these dependencies much stronger than others. In Fig. 1.1, each pixel in (a) is independent of every other pixel, and the lack of structure is evident. In (b) the pixels are only independent row-wise. Though the image still looks random, the structure is more apparent than that of (a). Finally, in (c) there is no independence betwen pixels and it seems much smoother than (a) and (b). Most natural images will have an overall apparent structure somewhere between (b) and (c) [18, 19]. This encourages the notion of reducing the complexity of our problem by assuming that pixels in an image have some structured dependency that can be approximately modeled and restricting our



(a) (b) (c)

Fig. 1.1: Various 300×300 ($p = 90000$) images: (a) every pixel independent, (b) pixels independent row-wise, (c) for $i = 1, \ldots, 90000$, pixel i generated as $[\sin(2\pi i/90000) + 1]/2$.

attention to carrying out analyses that respect this structure.

1.1.4.1 Structure Through Preprocessing

If we are given r landmark points in $\mathbb{R}^2$ contained in a matrix $\mathbf{A} \in \mathbb{R}^{r\times 2}$, affine transformations of shape can be obtained by right-multiplication of $\mathbf{A}$ by a 2×2 full rank matrix, say $\mathbf{B}$. Since $\mathbf{B}$ is full rank, span($\mathbf{A}$) will be invariant to right multiplication by $\mathbf{B}$ and thus invariant to affine transformations. After normalization through a singular value decomposition so that $\mathbf{A}^T\mathbf{A} = \mathbf{I}_2$, each set of landmark points will lie on $\mathcal{G}(r, 2)$ – that is, the space of all 2-dimensional linear subspaces of $\mathbb{R}^r$ [20, 21].

The FG-NET database is a typical source for benchmarking tasks such as age estimation. The database consists of 1002 images of individuals' faces, as well as landmark points for each individual. Additionally, attributes such as an individual's age or gender are given for each observation. See Fig. 1.2 for examples.

For this dataset, 68 predefined landmark points are given for each image in $\mathbb{R}^2$

resulting in each predictor $\mathbf{x} \in \mathbb{R}^{68\times 2}$. Normalizing the predictors to remove all affine transformations by performing a singular value decomposition on each observation is a useful preprocessing step resulting in $\mathbf{x} \in \mathcal{G}(68, 2)$. Predictors were concatenated column-wise to obtain vectors $\mathbf{x}_1, \ldots, \mathbf{x}_{1002} \in \mathbb{R}^{136}$.

Age estimation is a popular problem in computer vision that has seen numerous solutions that assume the predictors lie on a manifold. In [22], the low-dimensional representations for images labeled with an individual's age were used as predictors in a regression on the age of individuals in unlabeled images. Predictors were first embedded onto a lower-dimensional manifold and regression was performed on these transformed data points. An issue with this method is that it is difficult to know if any relevant information has been discarded.

Dynamic modeling for video analysis is another popular framework for inference in computer vision. As with the landmark points above, a similar preprocessing can be done when observations are video sequences. We assume to have, for each observation, a video sequence $\mathbf{x} \in \mathbb{R}^{r\times c\times T}$ where r is the number of rows in one frame, c the number of columns, and T the number of frames.

Here, an apperance model can be obtained by taking each observation (in this case a video sequence) and concatenating each frame of image vectors into a large matrix $\mathbf{A} \in \mathbb{R}^{rc\times T}$. We take the singular value decomposition of $\mathbf{A}$ to obtain an appearance matrix in $\mathbb{R}^{rc\times s}$ where s is a parameter to be chosen [23]. This approach is useful as it can greatly reduce the dimensionality of a video sequence, thus similarly reducing the computational cost.

Finally, domain adaptation [24] is a common problem in many computer vision

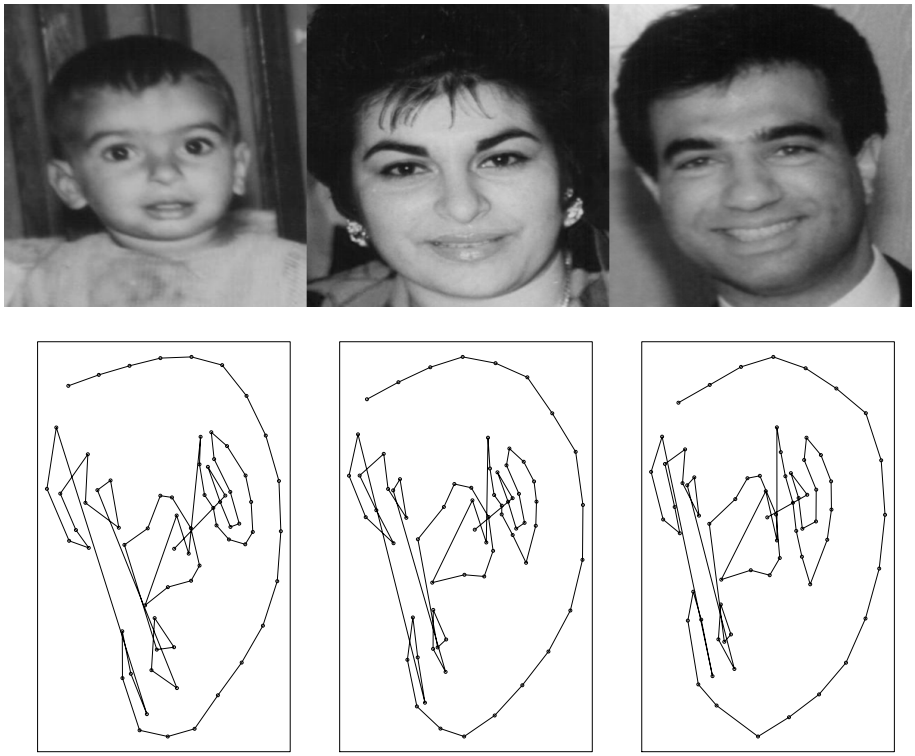


Fig. 1.2: Sample images from FG-NET originals (top) and landmark points (bottom). Images taken from [3].

problems. In it, we assume the predictors used for training – that is, for estimating model parameters – have a different distribution than the predictors used for testing – that is, for reporting prediction error using the model parameters obtained. Many approaches have been proposed to overcome these issues and are presented in Chapter 2.

1.2 Main Contributions

The main contributions of the dissertation are in the problems of regression on manifolds and in domain adaptation. For regression on manifolds, we propose a method similar to the exterior derivative estimator outlined in Chapter 2 that

deals with predictors lying on the Grassmannian. In this case, since the underlying structure of the predictors is known, the least squares objective can be modified with a projection that exploits this structure. This method is computationally efficient and can yield improvements in many computer vision problems.

For domain adaptation, we propose three related techniques. The first technique uses the information contained in both the training and testing data to obtain a dimension reduction of the predictors. This method can be extended to both include information about the response variable as well as be extended to cases in which the predictors lie on the Grassmannian. The second technique proposes a more well-defined method for incorporating information about the response through investigating an inverse regression approach; this method can also be extended to Grassmannian data through using additional transformations. Finally, the third technique seeks to extend the above via Monte Carlo methods in an attempt to incorporate both information about the conditional distribution of the response given the predictors as well as more complicated regularization functions.

1.3 Organization

This dissertation is organized as follows. Chapter 2 reviews some background on the problems considered. The first section outlines techniques for performing regression on high-dimensional predictors that have a low-dimensional, yet unknown structure. The second section presents various methods to handle data that may have a discrepancy between training and testing phases. Chapter 3 proposes a regu-

larized least squares method to handle high-dimensional predictors that come from a known underlying structure, and this method is applied to various regression and classification problems in computer vision. Chapter 4 talks about a dimension reduction method that can both incorporate information about the response as well as improve prediction in situations in which we have an inhomogeneity between the distributions of training and testing data. Chapter 5 proposes a dimension reduction method similar to that of Chapter 4 through posing a penalized maximum likelihood problem on the distribution of the predictors with the goal of still incorporating response information. Chapter 6 extends the dimension reduction method of Chapter 5 using sequential Monte Carlo while including the conditional model of the response given the predictors as well as the marginal distribution of the predictors. Finally, Chapter 7 summarizes the results obtained and indicates avenues for future research.

2

Background

2.1 Regression on Manifolds

As seen in the previous chapter, a common thread runs through high-dimensional inference problems: while the predictors have a large dimensionality, they also have an inherent structure that could cause problems when making predictions. This issue due to the structure of the predictors is often called *collinearity* or *near-collinearity* and is well-known in many areas of statistics. In this chapter, we investigate various techniques used to overcome the collinearities found in regression problems.

We assume predictors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are independent and distributed in $\mathcal{X} \subset \mathbb{R}^p$ and that the collinearity arises from the fact that $\mathcal{X}$ is a manifold. The response variables $y_1, \ldots, y_n \in \mathbb{R}$ are assumed to satisfy for each $i = 1, \ldots, n$

$$y_i = m(\mathbf{x}_\mathrm{i}) + \sigma(\mathbf{x}_\mathrm{i}) \cdot \epsilon_i$$

with ϵ_i independent and identically distributed (i.i.d.) where $E(\epsilon_i) = 0$ and $\mathrm{Var}(\epsilon_i) = 1$. Interest often lies in finding the regression function m at a point $\mathbf{x}_0$ defined as

$$m(\mathbf{x}_0) = E[Y \,|\, \mathbf{X} = \mathbf{x}_0]$$

where we take $(Y, \mathbf{X})$ to be the random variables associated with the joint distribution of the response and predictors, respectively.

If we assume $\dim(\mathcal{X}) = d$, exploiting the structure of $\mathcal{X}$ can aid matters if we take $d \ll p$, for example improving nonparametric convergence rates [25]. This manifold assumption can simplify matters on the theoretical level, but we still are tasked with finding an embedding into $\mathcal{X}$ as well as taking into account the manifold structure when performing analyses on the lower-dimensional data. However, if we are interested in predicting values in the high-dimensional space, operating on the low-dimensional representation and attempting to extrapolate this to the ambient space can result in difficulties. We seek a method that does not require knowledge of $\mathcal{X}$ explicitly, and thus will not rely upon an embedding.

2.1.1 Penalized Least Squares

We seek to estimate the regression function m in high-dimensional settings. Models that handle the presence of collinearities are helpful, but other properties are desired: interpretability, often achieved by obtaining a sparse model; the ability to handle the case in which $p \gg n$; automatic variable selection, which also helps to obtain a sparse model; and the explicit consideration of the underlying manifold structure.

Ordinary least squares (OLS) works by assuming a linear dependence of $\mathbf{X}$ on Y. In other words, solutions of the form $m(\mathbf{x}_0) = \beta_0 + \mathbf{x}_0^T \boldsymbol{\beta}_1$ are sought to minimize a the residual sum of squares

$$\hat{\boldsymbol{\beta}}_O = \arg\min_{\boldsymbol{\beta}} || \, \mathbf{y} - \mathbb{X} \, \boldsymbol{\beta} \, ||^2$$

where $\mathbf{y} = [y_1, \ldots, y_n]^T$ and

$$\mathbb{X} = \begin{bmatrix} 1 & \mathbf{x}_1^T \\ \vdots & \vdots \\ 1 & \mathbf{x}_\mathrm{n}^T \end{bmatrix}.$$

Here we note that $\mathbb{X}$ is slightly different than that defined in Chapter 1 as we wish to include an intercept in our model. We let $\boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}_1) \in \mathbb{R} \times \mathbb{R}^p$ and see that the solution can be obtained by solving the normal equation

$$\mathbb{X}^T \, \mathbb{X} \, \hat{\boldsymbol{\beta}}_O = \mathbb{X}^T \, \mathbf{y} \, .$$

The presence of collinearities in the predictors $\mathbf{X}$ causes issues with the rank of the matrix $\mathbb{X}^T \, \mathbb{X}$, indicating the need for more sophisticated methods. One such method to mitigate this effect is to add a regularization parameter to ensure that $\mathbb{X}^T \, \mathbb{X}$ is nonsingular. This can be incorporated into the minimization above by adding an ℓ_2 penalty on the parameters $\boldsymbol{\beta}$ and solving

$$\hat{\boldsymbol{\beta}}_R = \arg\min_{\boldsymbol{\beta}} || \, \mathbf{y} - \mathbb{X} \, \boldsymbol{\beta} \, ||_2^2 + \lambda || \, \boldsymbol{\beta}_1 \, ||_2^2$$

for $\lambda > 0$ a chosen parameter. We see now that

$$\hat{\boldsymbol{\beta}}_R = (\mathbb{X}^T\,\mathbb{X} + \lambda\,\mathbf{Q})^{-1}\,\mathbb{X}^T\,\mathbf{y}$$

where $\mathbf{Q} = \mathrm{diag}[0, \mathbf{I}_\mathrm{p}]$, and the singularity issues are no longer present. This method – known as *ridge regression* [26] – can also handle the case in which $p \gg n$ due to the regularization performed on the parameter vector $\boldsymbol{\beta}$.

A straightforward extension of ridge regression is the least absolute shrinkage and selection operator (LASSO) which employs an ℓ_1 penalty as opposed to an ℓ_2 penalty on the parameters, i.e.,

$$\hat{\boldsymbol{\beta}}_L = \arg\min_{\boldsymbol{\beta}} ||\,\mathbf{y} - \mathbb{X}\,\boldsymbol{\beta}\,||_2^2 + \lambda||\,\boldsymbol{\beta}_1\,||_1$$

where $||\cdot||_1 = \sum_{j=1}^{p} |\beta_j|$. Though there is no closed form solution in general, if $\mathbb{X}$ is an orthogonal matrix and the $\mathbf{x}$ values are centered, then it can be shown that

$$\hat{\boldsymbol{\beta}}_{1L} = \mathrm{sgn}(\hat{\boldsymbol{\beta}}_{1O})(|\hat{\boldsymbol{\beta}}_{1O}| - \tilde{\lambda})^+$$

where $\hat{\boldsymbol{\beta}}_{1O}$ is the OLS solution, $(\cdot)^+$ denotes the positive part of its input and $\tilde{\lambda}$ is determined by the constraint involving $\lambda||\,\boldsymbol{\beta}_1\,||_1$. Thus for large enough λ the LASSO solution has the benefit that certain elements of $\hat{\boldsymbol{\beta}}_{1L}$ will be exactly zero resulting both in a sparse model and automatic variable selection. A further extension of both ridge regression and the LASSO is the elastic net (EN) [27] which combines both an ℓ_1 and an ℓ_2 penalty into the regularization resulting in the optimization seeking

$$\hat{\boldsymbol{\beta}}_{EN} = \arg\min_{\boldsymbol{\beta}} || \mathbf{y} - \mathbb{X}\boldsymbol{\beta} ||_2^2 + \lambda[\alpha|| \boldsymbol{\beta}_1 ||_1 + (1-\alpha)|| \boldsymbol{\beta}_1 ||_2^2].$$

The elastic net enjoys the benefits of both ridge regression and the LASSO: like ridge regression it is able to adequately account for collinearities in the predictors and can handle the case of $p \gg n$; like the LASSO it is able to give a sparse model and can perform automatic variable selection.

A slight drawback to the above methods is that, under the assumption that predictors $\mathbf{x}_i$ lie on the manifold $\mathcal{X}$, none of the methods explicitly consider this underlying structure. Removing the effect of collinearities on the predictors via principal components regression (PCR) [28] attempts to exploit this underlying structure. Principal components regression works by finding the d largest principal components of the covariance matrix for $\mathbb{X}$ and performing regression on the transformed variables. The benefit of PCR can be largely problem-dependent as the handling of collinearities and the problem of $p \gg n$ will depend on how many principal components are included in the final regression. By design PCR gives a sparse model – in fact, the model obtained is as "sparse" as the practitioner desires as it will only contain d predictors. However, each new predictor is a linear combination of all of the original predictors, which does not indicate sparsity in the ambient space. Principal components regression explicitly considers the manifold structure in the predictors by only keeping projections of the variables that have the highest variation, though this only applies to the case in which $\mathcal{X}$ is globally linear.

2.1.2 Local Regression

Formally, the "curse of dimensionality" in nonparametric statistics described in Chapter 1 concerns the fact that using nonparametric regression to estimate the function m breaks down as the dimension of $\mathbf{X}$ grows large. It can be shown [6] that if the true regression function m has smoothness of order ρ then no nonparametric estimator of m will have a faster convergence rate for the root mean integrated square error (RMISE) than $n^{-\rho/(2\rho+p)}$. Typically $\rho \leq 2$ in practice; estimates of the regression function $m(\mathbf{x})$ for $\rho > 2$ will achieve better rates of convergence, but we will still be left with poor convergence rates for large p, as well as problems of overfitting.

Here we outline a few nonparametric estimates of the regression function m. For each $(y_i, \mathbf{x}_\mathrm{i})$, the ordinary least squares solution described in Section 2.1.1 can be derived through minimizing the sum of squared errors of the model

$$y_i = \beta_0 + \mathbf{x}_\mathrm{i}^T \boldsymbol{\beta}_1 + \epsilon_i$$

for ϵ_i a random error term. To obtain a better fit to the data and incorporate nonlinearities and local structure into the model, we consider a local regression about the point $\mathbf{x}_0$. In this case

$$y_i = \beta_0(\mathbf{x}_0) + \mathbf{x}_\mathrm{i}^T \boldsymbol{\beta}_1(\mathbf{x}_0) + \epsilon_i, \text{ for } \mathbf{x}_\mathrm{i} \in \mathbf{U}_h(\mathbf{x}_0) \tag{2.1}$$

for $\mathbf{U}$ a local neighborhood about $\mathbf{x}_0$ and h its size, and the parameters $\boldsymbol{\beta}$ depend on

the location of $\mathbf{x}_0$. One solution to the problem of estimating the regression function m is given by the Nadaraya-Watson estimator [29]

$$\hat{m}_h(\mathbf{x}_0) = \frac{\sum_{i=1}^{n} K_h(\mathbf{x}_\mathrm{i} - \mathbf{x}_0) \cdot y_i}{\sum_{i=1}^{n} K_h(\mathbf{x}_\mathrm{i} - \mathbf{x}_0)}$$

where K_h is a weighting kernel $K_h(\mathbf{u}) = h^{-p}K(\mathbf{u}/h)$. Note, however, that the Nadaraya-Watson estimator fits a local intercept, ignoring the linear term $\mathbf{x}_\mathrm{i}^T \boldsymbol{\beta}_1(\mathbf{x}_0)$ in (2.1). Often restrictions are placed on K, such as requiring it to be continuous, radially symmetric, and integrate to one. For high-dimensional data, a popular kernel to use is the radial Gaussian kernel where

$$K(\mathbf{u}) = \exp\left[-\frac{1}{2}||\,\mathbf{u}\,||^2\right].$$

This kernel can be generalized by considering a bandwidth matrix $\mathbf{H}$ as opposed to the same bandwidth h for each component, which does not necessarily have radial symmetry.

The random term in the denominator of the Nadaraya-Watson estimator is not always desirable, and the Gasser-Müller estimator [30] attempts to overcome this issue. The estimator is given by

$$\hat{m}_h(\mathbf{x}_0) = y_i \sum_{i=1}^{n} \int_{\mathbf{s}_\mathrm{i}}^{\mathbf{s}_{\mathrm{i}+1}} K_h(\mathbf{u} - \mathbf{x}_0)\; d\,\mathbf{u}$$

where $\mathbf{s}_\mathrm{i} = (\mathbf{x}_\mathrm{i} + \mathbf{x}_{i+1})/2$, $\mathbf{x}_0 = -\infty$ and $\mathbf{x}_{n+1} = +\infty$.

The local linear estimator [31] attempts to solve the local regression problem

with the benefit of being able to be posed as a least squares problem. If we form the weight matrix and augmented data matrix as

$$\mathbf{W}_{\mathrm{x0}} = \mathrm{diag}[K_h(\mathbf{x}_1 - \mathbf{x}_0), \ldots, K_h(\mathbf{x}_\mathrm{n} - \mathbf{x}_0)],\ \mathbb{X}_{\mathrm{x0}} = \begin{bmatrix} 1 & (\mathbf{x}_1 - \mathbf{x}_0)^T \\ \vdots & \vdots \\ 1 & (\mathbf{x}_\mathrm{n} - \mathbf{x}_0)^T \end{bmatrix},$$

the local linear regression estimator solves

$$\hat{\boldsymbol{\beta}}(\mathbf{x}_0) = \arg\min_{\boldsymbol{\beta}} = || \mathbf{W}_{\mathrm{x0}}^{1/2}(\mathbf{y} - \mathbb{X}_{\mathrm{x0}}\,\boldsymbol{\beta})||^2, \tag{2.2}$$

yielding estimates

$$\hat{m_h}(\mathbf{x}_0) = \hat{\beta}_0,\ \widehat{\frac{\partial m_h}{\partial\,\mathbf{x}_\mathrm{j}}}(\mathbf{x}_0) = \hat{\beta}_j,\ j = 1, \ldots, p.$$

The main problem in local linear regression is choice of the optimal bandwidth h, typically using the mean integrated square error (MISE) as a criterion. To obtain the MISE, we note that there exist functions J_1 and J_2 such that [31]

$$E(\hat{m}_h(\mathbf{x}_0) - m(\mathbf{x}_0)|\,\mathbf{x}_1, \ldots, \mathbf{x}_\mathrm{n}) = h^2 J_1(\mathbf{x}_0)(1 + o_P(1)),$$

$$\mathrm{var}(\hat{m}_h(\mathbf{x}_0) - m(\mathbf{x}_0)|\,\mathbf{x}_1, \ldots, \mathbf{x}_\mathrm{n}) = n^{-1}h^{-p} J_2(\mathbf{x}_0)(1 + o_P(1)),$$

where $o_P(\cdot)$ denotes order in probability pointwise. Now

$$\text{MISE}(h) = \int \left\{ [h^2 J_1(\mathbf{u})(1 + o_P(1))]^2 + n^{-1} h^{-p} J_2(\mathbf{u})(1 + o_P(1)) \right\} \, d\,\mathbf{u}$$

yielding an optimal convergence rate of $n^{-4/(4+p)}$. Since we assume p is large for high-dimensional problems, this rate will lead to poor convergence.

The main result of [25] concerns the behavior of both the bias and variance of $\hat{m}_h(\mathbf{x}_0)$ when we consider $\mathcal{X}$ to be a manifold of dimension d. Under certain assumptions – mostly on the kernel K and the manifold $\mathcal{X}$ – it can be shown that if $\mathbf{x}_0$ is an interior point of $\mathcal{X}$, there exist some $J_1(\mathbf{x}_0)$ and $J_2(\mathbf{x}_0)$ such that

$$E(\hat{m}_h(\mathbf{x}_0) - m(\mathbf{x}_0) | \, \mathbf{x}_1, \ldots, \mathbf{x}_\text{n}) = h^2 J_1(\mathbf{x}_0)(1 + o_P(1)),$$

$$\text{var}(\hat{m}_h(\mathbf{x}_0) - m(\mathbf{x}_0) | \, \mathbf{x}_1, \ldots, \mathbf{x}_\text{n}) = n^{-1} h^{-d} J_2(\mathbf{x}_0)(1 + o_P(1)).$$

The usefulness of this result is twofold: a faster optimal rate of $n^{-4/(4+d)}$ can be obtained for the conditional MISE of $\hat{m}(x_0)$ by choosing $h = \kappa n^{-1/(4+d)}$ for some $\kappa > 0$ where now $d \ll p$ is the dimension of the manifold $\mathcal{X}$; additionally, this method does not require points to be embedded into a lower-dimensional manifold which can often be too restrictive in practice. Unfortunately, this method still leaves the task of estimating κ as well as the dimension d from the data which can be nontrivial. Another drawback of this method is the introduction of problems in the rank of the local covariance matrix which is used to find the optimal parameters $\hat{\boldsymbol{\beta}}$. The solution to (2.2) is

$$\hat{m}(\mathbf{x}_0) = \mathbf{e}_1^{\mathrm{T}} (\mathbb{X}_{\mathrm{x0}}^T \, \mathbf{W}_{\mathrm{x0}} \, \mathbb{X}_{\mathrm{x0}})^{-1} \, \mathbb{X}_{\mathrm{x0}}^T \, \mathbf{W}_{\mathrm{x0}} \, \mathbf{y}$$

where $\mathbf{e}_1$ is the first column of the matrix $\mathbf{I}_{p+1}$. This solution depends on the inverse of the local covariance matrix $\mathbb{X}_{\mathrm{x0}}^T \, \mathbf{W}_{\mathrm{x0}} \, \mathbb{X}_{\mathrm{x0}}$. It is often the case that this covariance is ill-conditioned due to the structure of the underlying manifold $\mathcal{X}$.

2.1.3 The Exterior Derivative Estimator

The exterior derivative estimator (EDE, [9]) seeks

$$\hat{\boldsymbol{\beta}}_{EDE} = \arg\min_{\boldsymbol{\beta}} || \, \mathbf{y} - \mathbb{X} \, \boldsymbol{\beta} \, ||_2^2 + \lambda || \, \boldsymbol{\Pi} \, \boldsymbol{\beta}_1 \, ||_2^2$$

for $\boldsymbol{\Pi}$ a projection matrix. The projection matrix $\boldsymbol{\Pi}$ is included to consider the manifold structure $\mathcal{X}$ by penalizing the coefficients $\boldsymbol{\beta}$ for not falling onto the directions of the manifold. Since this will only take into consideration manifolds that are globally linear, a modification to the minimization above is done by weighting each observation as in Section 2.1.2 [this is known as the nonparametric exterior derivative estimator (NEDE)]. The projection is done by penalizing the coefficient vector in directions perpendicular to the tangent space formed by the data points $\mathbf{x}_1, \ldots, \mathbf{x}_{\mathrm{n}}$. In other words, let

$$\mathbf{H}^{1/2} = \sqrt{nh^d} \begin{bmatrix} 1 & 0 \\ 0 & h \cdot \mathbf{I}_{\mathrm{p}} \end{bmatrix},$$

and

$$\hat{\mathbf{C}}_n := \begin{bmatrix} \hat{c}_n^{11} & \hat{\mathbf{c}}_n^{12} \\ \hat{\mathbf{c}}_n^{21} & \hat{\mathbf{C}}_n^{22} \end{bmatrix} = h^p \cdot \mathbf{H}^{-1/2} \, \mathbb{X}_{\mathrm{x}0}^T \, \mathbf{W}_{\mathrm{x}0} \, \mathbb{X}_{\mathrm{x}0} \, \mathbf{H}^{-1/2},$$

which acts as a matrix that captures the local covariance structure of the predictors. As is done in principal component analysis (PCA), we perform an eigenvalue decomposition of $\hat{\mathbf{C}}_n^{22}$

$$\hat{\mathbf{C}}_n^{22} = [\hat{\mathbf{R}} \; \hat{\mathbf{N}}] \cdot \hat{\mathbf{\Lambda}} \cdot [\hat{\mathbf{R}} \; \hat{\mathbf{N}}]^T$$

where $\hat{\mathbf{R}} \in \mathbb{R}^{p \times d}, \hat{\mathbf{N}} \in \mathbb{R}^{p \times (p-d)}$ and $\mathbf{\Lambda} \in \mathbb{R}^{p \times p}$ is a diagonal matrix with nonincreasing entries on the diagonal. Given this decomposition we construct the projection matrix and regularization matrix as

$$\hat{\mathbf{\Pi}}_{\mathbf{x},0} := \hat{\mathbf{N}}\hat{\mathbf{N}}^T, \;\; \hat{\mathbf{P}}_n := \mathrm{diag}(0, \hat{\mathbf{\Pi}}_{\mathbf{x},0}).$$

With this, we see

$$\hat{\boldsymbol{\beta}}_{EDE} = (\mathbb{X}^T \mathbb{X} + \lambda \hat{\mathbf{P}}_n)^{-1} \mathbb{X}^T \mathbf{y}$$

is the EDE solution.

A drawback of the above solution is that it performs poorly in the event that $p \gg n$ since the eigenvectors and eigenvalues of $\hat{\mathbf{C}}_n$ are not guaranteed to converge to their true values [9]. We can overcome these problems by regularizing the covariance and cross-covariance matrices by introducing a thresholding operator

$$T_t(\mathbf{M}) = \{m_{ij} \cdot \mathbf{1}(|m_{ij}| \geq t)\}$$

where $\mathbf{M} = \{m_{ij}\}$ is a matrix. If the true covariance matrix is adequately sparse then this hard-thresholding of each element of the estimated covariance matrix (between $\mathbb{X}_{\text{x0}}$ and $\mathbb{X}_{\text{x0}}$) and cross-covariance matrix (between $\mathbb{X}_{\text{x0}}$ and $\mathbf{y}$) will give consistent estimates to the true covariance and cross-covariance matrix and can be used in solving the weighted least squares problem [32, 9].

Due to the need for fitting a different model at every test data point, high computational costs are a serious drawback to this method of regression. In this respect, this approach is similar in spirit to computer vision algorithms that require dimension reduction on images as training points using some form of image differencing [33].

2.1.4 Extension to Classification Problems

All of the previously outlined estimators were concerned with the case in which the response variable was continuous. A method for converting these estimators into ones that can take into account categorical response variables uses regularized logistic regression. For a binary response ($y_i \in \{+1, -1\}$), the *logit* link function is used and the linear dependence on the parameters and predictors is through this function, i.e.,

$$\log \frac{\mathrm{P}(y_i = +1 |\, \mathbf{x}_\mathrm{i})}{\mathrm{P}(y_i = -1 |\, \mathbf{x}_\mathrm{i})} = \beta_0 + \mathbf{x}_\mathrm{i}^T \boldsymbol{\beta}_1 \,.$$

Instead of minimizing a regularized ℓ_2 loss function on the error in the model, the concern now is to maximize a penalized likelihood function, which amounts to solving

$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \left\{ L(\boldsymbol{\beta}) - \lambda ||\hat{\mathbf{P}}_n \boldsymbol{\beta}||^2 \right\}$$

where

$$L(\boldsymbol{\beta}) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(y_i = +1) \log p(\mathbf{x}_\mathrm{i}) + \mathbf{1}(y_i = -1) \log(1 - p(\mathbf{x}_\mathrm{i})).$$

This can be accomplished by an application of the Newton-Raphson algorithm to give an iteratively reweighted least squares problem [34]. We have a quadratic approximation to the log-likelihood function

$$L(\boldsymbol{\beta}) = -\frac{1}{2n} \sum_{i=1}^{n} w_i (\mathbf{z}_\mathrm{i} - \beta_0 - \mathbf{x}_\mathrm{i}^T \boldsymbol{\beta}_1)^2 + C(\tilde{\boldsymbol{\beta}})$$

where $\tilde{\boldsymbol{\beta}}$ are the current estimated parameters,

$$\mathbf{z}_\mathrm{i} = \tilde{\beta}_0 + \mathbf{x}_\mathrm{i}^T \tilde{\boldsymbol{\beta}}_1 + \frac{y_i - \tilde{p}(\mathbf{x}_\mathrm{i})}{\tilde{p}(\mathbf{x}_\mathrm{i})(1 - \tilde{p}(\mathbf{x}_\mathrm{i}))}$$

is the working response and

$$w_i = \tilde{p}(\mathbf{x}_\mathrm{i})(1 - \tilde{p}(\mathbf{x}_\mathrm{i}))$$

where $\tilde{p}(\mathbf{x}_\mathrm{i})$ is evaluated at the current parameters $\tilde{\boldsymbol{\beta}}$ (see [35] and [36] for detailed

derivations). The goal is to find

$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \left\{ \ell(\boldsymbol{\beta}) - \lambda ||\hat{\mathbf{P}}_n \boldsymbol{\beta}||^2 \right\},$$

which can be solved using various methods (iteratively reweighted least squares [37], coordinate descent [38]).

2.1.5 An Illustrative Example

The technique of weighted, regularized least squares can be applied to an array of problems in pattern recognition, especially in the field of computer vision. A simple example was given in [9] in which horizontal pose angles of images were estimated using various views of an artificial face from the ISOMAP face dataset [15]. In an example in [39], models using ordinary differential equations are used for system identification, and the predictors used are similarly highly correlated. We compare various methods applied to the FG-NET database [3] described in Chapter 1.

Each image is converted to normalized grayscale taking values between 0 and 1. On each image, we use the Viola-Jones face detection algorithm [40] to discard much of the noise and unwanted information contained in the background. Finally, we rescale each image to a size of 25×25 pixels. It is argued in [41] that this last step is justified due to the assumption that the predictors lie on a lower-dimensional manifold, and thus a uniform resizing of the images will not lead to a loss of information in the predictors.

The final data we analyze consist of 1002 vectors of size 625, each labeled with ages ranging from 0 to 69. In order to investigate the utility of some of these techniques in a classification setting, we consider the indicator of whether or not an individual's age was less than 21 as a response variable. We adapt each of the methods of regression to perform logistic regression on these binary response variables as described in Section 2.1.4. The three alternative methods are the elastic net (EN), principal components regression (PCR), and regression on points embedded using locality preserving projections (LPP) [42]. The last two methods were suggested in [22], though it was mentioned that PCR will most likely not perform well due to the fact that it does not perform an embedding that is able to discriminate classes well. All forms of regression are performed on either the predictors or the embedded predictors; including the square of the predictors as well has been shown to aid in prediction [22], but this is not considered in this analysis.

A few issues need to be taken care of when implementing the described methods. The dependence of the regularization on the dimension d of the underlying manifold is a major problem which we must solve using methods of estimating data dimension given a finite number of samples. Many such techniques are available in the machine learning literature [43]. A popular technique uses maximum likelihood [44]. Other methods, such as those using nearest neighbor or principal component analysis, can be used. Whatever method is used, we consider the dimension fixed once it is estimated at a point. Note, however, that the local regression approach has the added benefit that the dimension of the manifold is free to differ point-to-point, giving flexibility over methods that require a strict embedding of the points into a

lower-dimensional space.

A difficult problem in applications is the selection of tuning parameters. Many of the models we consider have multiple tuning parameters to estimate which can lead to possible over- or under-fitting if they are not chosen properly. We take $\hat{d} = 17$ and $\hat{\lambda} = 0.1$ for the EDE. We do not perform local regression, thus avoiding the issue of choosing a kernel and bandwidth. For EN, we take α to be 1 and λ as 10^{-5}, resulting in a mild LASSO penalty. For LPP, a sufficiently large $\hat{d}$ of 300 yields positive results, while for PCA we take $\hat{d}$ as 17 for comparison to the other methods. As is shown in Fig. 2.1, the EDE performs well compared with the alternative methods that were considered on the full data.

2.2 Prediction Using Inhomogeneous Data

In many statistical problems, we typically assume homogeneity of distributions between training and testing. In other words, parameter selection and error reporting are typically done by assuming we have a certain amount of training data $\mathbf{x}_1, \ldots, \mathbf{x}_n$ to estimate model parameters with additional data $\mathbf{x}_1^*, \ldots, \mathbf{x}_m^*$ to use for testing. For example, we could use $\mathbf{x}_i^*$ for selecting optimal parameters by finding those parameters that minimize some cross validation criterion; we could similarly use these test data to report how well a method can predict response values for unseen data. In all cases previously considered we have assumed that both data $\mathbf{x}_i$ and $\mathbf{x}_j^*$ are distributed similarly, namely as the random variable $\mathbf{X}$. Often in practical situations the disbtributions between the training and testing phases will not be

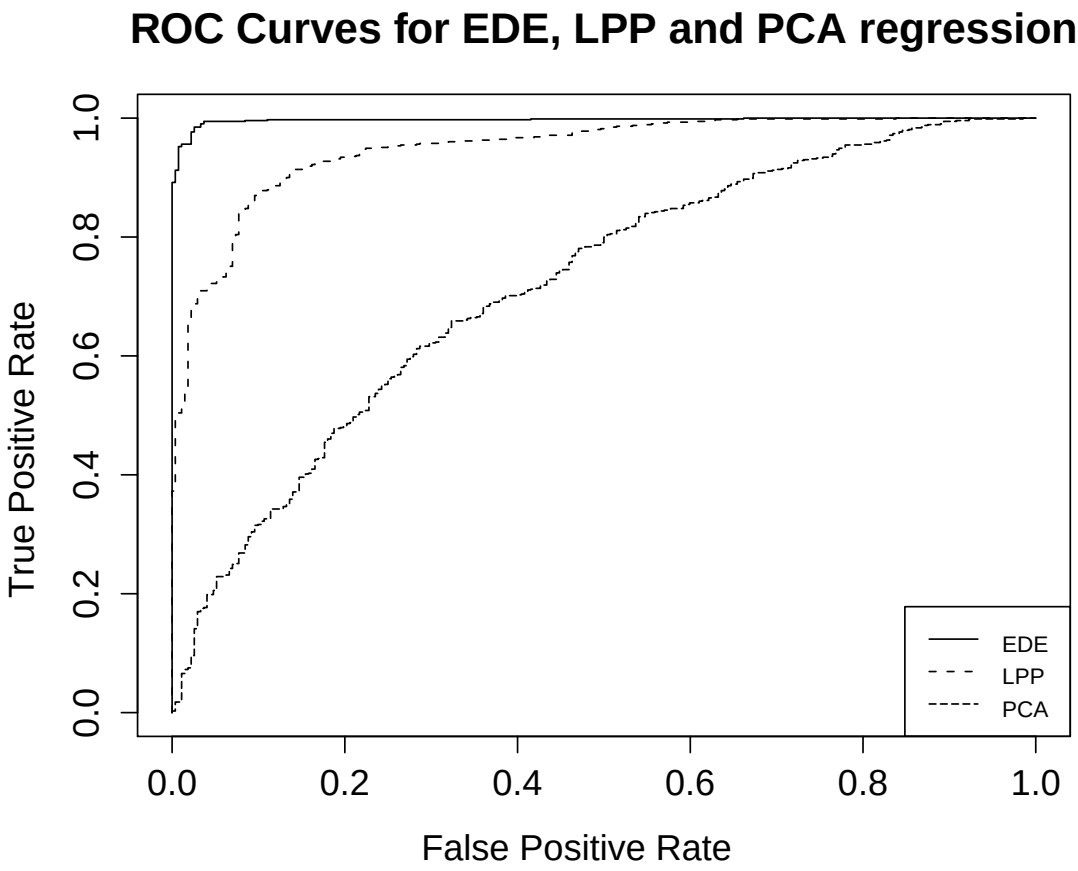


Fig. 2.1: ROC curve for exterior derivative estimator (EDE) and regression after dimension reduction via locality preserving projections (LPP) and after principal component analysis (PCA).

homogeneous, which can can result in poor predictive performance and present difficulties in determining optimal tuning parameters through cross-validation.

We formalize this problem by operating under the assumption that the joint distribution between the response variable (Y) and covariates ($\mathbf{X}$) changes from training to testing. In domain adaptation [24], we assume that while the covariate distribution might change between two domains, the underlying mechanism to generate the response variables from the covariates does not change. In other words, the conditional distribution of the response given the covariates remains the same across all domains, while the marginal distribution of the covariates may change. These assumptions are also present in covariate shift problems [45]. Transfer learning, another approach to this problem, deals with the problem of the conditional distribution of the response given the covariates changing between domains while the covariate distribution stays the same [46].

We assume to have independent data $\mathbf{x}_1, \ldots, \mathbf{x}_n$ with $\mathbf{x}_i \sim \mathbf{X} \in \mathcal{X} \subset \mathbb{R}^p$ with corresponding response variables $y_1, \ldots, y_n \in \mathcal{Y}$ for training a model (called the "source" data) and $\mathbf{z}_1, \ldots, \mathbf{z}_m$ with $\mathbf{z}_j \sim \mathbf{Z} \in \mathcal{Z} \subset \mathbb{R}^q$ with corresponding response variables $\xi_1, \ldots, \xi_m$ for testing (called the "target data"). Our main assumption is that we have unknown response variables $\xi_1, \ldots, \xi_m$ from the same model that generated the known response variables (i.e., $[Y \,|\, \mathbf{X}] \sim [\Xi \,|\, \mathbf{Z}]$, though $\mathbf{X} \nsim \mathbf{Z}$ in general). Our goal is to learn a parameterized conditional model optimal under $(\Xi, \mathbf{Z})$ while only knowing a small number of observations from Ξ (or knowing none in the unsupervised case). Our data will typically consist of $\mathbb{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times p}$, $\mathbf{y} = [y_1, \ldots, y_n]^T \in \mathbb{R}^n$, and $\mathbb{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_m]^T \in \mathbb{R}^{m \times q}$. On occasion we will

be blessed with some labeled examples from the target data, though all methods will be outlined for the unsupervised case, with the semisupervised case being a straightforward extension.

Common solutions to problems dealing with discrepancies between training and testing data have been sought in a variety of fields. Survey statistics is possibly one of the oldest fields to attempt to answer questions regarding the effect a difference in distribution has on the utility of models and the inferences they can make. The most prevalent solution to this problem in survey statistics is to seek a weighting of each of the observed data points in order to accurately extrapolate the information present in the given data to unseen data points. See [47] for a thorough introduction. Sample selection bias [48] – an approach from econometrics – describes the bias inherent in using nonrandomly selected data points to form models as a specification error, i.e., an error arising from the inconsistency between the initial model assumptions and the true nature of the sample.

2.2.1 Instance-Weighting Methods

Instance-weighting methods seek a set of weights to apply to the source data that will transform the distribution of the labeled source data into that of the unlabeled (or partially labeled) target data. This way, transporting a model from the labeled data in a different domain will yield hopefully better results. Shimodaira [45] proposed a method for correcting this discrepancy with a view toward improving predictive performance by weighting each element in the source data by an impor-

tance weight based on both the source and target density functions.

In covariate shift (CS, [45]), we assume the source data $\mathbf{X} \sim f$ and target data $\mathbf{Z} \sim g$ for some density functions f and g. If we operate under the previous assumption of equivalent conditional distributions, we fix a parameterization for the conditional distribution and define the Kullback-Leibler loss for the source data as

$$L_{\mathbf{x}}(\boldsymbol{\beta}) = -\int_{(\mathcal{Y},\mathcal{X})} f(\mathbf{x})\ q(y|\,\mathbf{x})\ \log p(y|\,\mathbf{x},\boldsymbol{\beta})\ dy\ d\,\mathbf{x}$$

and for the target data as

$$L_{\mathbf{z}}(\boldsymbol{\beta}) = -\int_{(\Xi,\mathcal{Z})} g(\mathbf{z})\ q(\xi|\,\mathbf{z})\ \log p(\xi|\,\mathbf{z},\boldsymbol{\beta})\ d\xi\ d\,\mathbf{z}\,.$$

We assume for the time being that $(Y, \mathbf{X})$ and $(\Xi, \mathbf{Z})$ have the same support so that the loss functions $L_{\mathbf{x}}$ and $L_{\mathbf{z}}$ differ only through the marginal distributions f and g. Let

$$L_w^{(n)}(\boldsymbol{\beta}; \mathbb{X}, \mathbf{y}) = -\sum_{i=1}^{n} w(\mathbf{x}_\mathrm{i})\ \log p(y_i|\,\mathbf{x}_\mathrm{i},\boldsymbol{\beta})$$

so that, for $w(\mathbf{x}) \equiv 1$, $L_w^{(n)}/n \to L_{\mathbf{x}}$ as $n \to \infty$. Since we desire the loss for the target data $\mathbf{Z}$, we take w to be the importance weights $w(\mathbf{x}) = g(\mathbf{x})/f(\mathbf{x})$. In this case, $L_w^{(n)}/n \to L_{\mathbf{z}}$ as $n \to \infty$, which is what is needed.

This method is not always optimal. First, it typically will only yield improvements under a misspecification of the model [i.e., in the case where $p(y|\,\mathbf{x},\boldsymbol{\beta})$ differs from the "true" model], though for high-dimensional data even standard methods for

misspecified models perform similarly to the importance-weighted method [49]. Additionally, density estimation is a nontrivial problem in high dimensions, which is often the case in problems of interest. Density estimates can still be made by, for example, using radial kernels, though accurate estimates of the density are still difficult in this case. Even for one-dimensional problems, CS requires that $\text{supp}(\mathcal{Z}) \subseteq \text{supp}(\mathcal{X})$, a restriction that can hinder the types of problems we wish to consider.

Kernel mean matching (KMM, [50]) seeks to generalize CS by incorporating the weights w into an objective function and optimizing over the data to find optimal weights instead of taking them as "known." CS is further generalized by defining a kernel function $\Phi : \mathcal{X} \to \mathcal{F}$ and assuming still that $\mathcal{X} = \mathcal{Z}$. An expectation operator μ is defined as

$$\mu(F) = E_F[\Phi(\mathbf{x})]$$

where F is the cdf corresponding to the density f. The KMM procedure seeks to solve

$$\underset{w}{\text{minimize}} \; ||\mu(G) - E_F[w(\mathbf{x})\Phi(\mathbf{x})]||$$

$$\text{subject to } w(\mathbf{x}) \geq 0 \text{ and } E_F[w(\mathbf{x})] = 1.$$

To find $\mathbf{w} \in \mathbb{R}^n$, we will define an empirical version of the above objective function and incorporate constraints $w_i \in [0, W]$ and $|n - \sum_{i=1}^n w_i| \leq n\epsilon$, analogous to the two constraints above. This empirical objective will be written as

$$J(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T \mathbf{K}\, \mathbf{w} - \mathbf{k}^T \mathbf{w}$$

where

$$\mathbf{K}_{ij} = k(\mathbf{x}_\mathrm{i}, \mathbf{x}_\mathrm{j}),\ \mathbf{k}_i = \frac{n}{m}\sum_{j=1}^{m} k(\mathbf{x}_\mathrm{i}, \mathbf{z}_\mathrm{j})$$

for an appropriate kernel function $k(\cdot, \cdot)$. The constrained minimization of J above using the empirical constraints can be solved by any quadratic programming optimization (e.g., interior point methods [51]). In [50], k is taken to be the Gaussian kernel

$$k(\mathbf{u}, \mathbf{v}) = \exp\{-\sigma || \mathbf{u} - \mathbf{v} ||^2\}$$

for a fixed σ.

Choice of ϵ will be governed by the following result. Huang, et. al [50] showed that if $w(\mathbf{x}) \in [0, W]$ for all $\mathbf{x} \in \mathcal{X}$, given $\mathbf{x}_1, \ldots, \mathbf{x}_n$ i.i.d. from F, as a direct consequence of the central limit theorem we have

$$\frac{1}{n}\sum_{i=1}^{n} w(\mathbf{x}_\mathrm{i}) \xrightarrow{D} N(\mu_w, \sigma_w^2)$$

where $\mu_w = \int_{\mathcal{X}} w(\mathbf{x})\, dF(\mathbf{x})$ and $\sigma_w \leq W/(2\sqrt{n})$. This indicates that we should take $\epsilon = O(W/\sqrt{n})$ for some fixed constant W.

KMM is able to overcome some of the drawbacks of CS. Since weights are estimated directly, density functions no longer need to be defined explicitly (or

estimated), reducing potential errors, especially in multivariate problems. Moreover, the kernel approach allows for capturing possible nonlinearities that might improve predictive performance. Unfortunately, we still have the restriction that the source and target must have the same support, and in fact still need $\text{supp}(\mathcal{Z}) \subseteq \text{supp}(\mathcal{X})$ as in CS. Additionally, since a kernel approach is used, weights are only defined for the training input points so that if we desire weights for points that were not available initially (e.g., for cross validation), we will need to rerun the optimization.

We see the Kullback-Leibler importance estimation procedure (KLIEP, [49]) as an attempt to improve on KMM by specifying a model for the weight function $w(\mathbf{x})$ so that weights can be obtained for points not available at training, instead estimating them using the given data. We wish to model

$$\hat{g}(\mathbf{x}) = \hat{w}(\mathbf{x}) \cdot f(\mathbf{x}), \;\; \hat{w}(\mathbf{x}) = \sum_{\ell=1}^{b} \alpha_\ell \phi_\ell(\mathbf{x})$$

where $\{\alpha_\ell\}$ are unknown parameters to be estimated and $\phi_\ell \geq 0$ are fixed, non-negative basis functions. The weight function $\hat{w}$ will be chosen to minimize the Kullback-Leibler divergence between g and $\hat{g}$, that is,

$$D_{KL}[g(\mathbf{x})||\hat{g}(\mathbf{x})] = \int_{\mathcal{X}} g(\mathbf{x}) \;\; \log \frac{g(\mathbf{x})}{f(\mathbf{x})} \; d\,\mathbf{x} - \int_{\mathcal{X}} g(\mathbf{x}) \;\; \log \hat{w}(\mathbf{x}) \; d\,\mathbf{x}\,.$$

Since the second term is the only one with the parameters of interest, we form the loss function as

$$J(\boldsymbol{\alpha}) = -\frac{1}{m}\sum_{j=1}^{m}\log\sum_{\ell=1}^{b}\alpha_\ell\phi_\ell(\mathbf{z}_\mathrm{j})$$

where b is chosen from the data. Unfortunately J is concave so we require constraints to successfully optimize over $\boldsymbol{\alpha}$. First, we desire $\hat{w}(\mathbf{x}) \geq 0$, which we specify as

$$\alpha_\ell \geq 0 \text{ for } \ell = 1, \ldots, b.$$

Additionally, since $\hat{g}(\mathbf{x}) = \hat{w}(\mathbf{x})f(\mathbf{x})$ should be a proper density function, we have

$$1 = \int_{\mathcal{X}} \hat{w}(\mathbf{x})\ f(\mathbf{x})\ d\,\mathbf{x} \approx \frac{1}{n}\sum_{i=1}^{n}\sum_{\ell=1}^{b}\alpha_\ell\phi_\ell(\mathbf{x}_\mathrm{i}).$$

In practice, we will define basis functions using the target data. Here,

$$\hat{w}(\mathbf{x}) = \sum_{j=1}^{m}\alpha_\ell k(\mathbf{x}, \mathbf{z}_\mathrm{j})$$

where $k(\cdot,\cdot)$ is the Gaussian kernel as before, taking b as m, that is, the number of basis functions is equal to the number of target data points.

The estimation of α will be done through gradient ascent of the negative of J above. We will define

$$\mathbf{K}_{ij} = k(\mathbf{z}_\mathrm{i}, \mathbf{z}_\mathrm{j}),\ \mathbf{k}_j = \frac{1}{n}\sum_{i=1}^{n}k(\mathbf{x}_\mathrm{i}, \mathbf{z}_\mathrm{j})$$

as in KMM, except this time with the target data instead of the source data. In this case, the parameter update uses gradient ascent with $\mathbf{K}$ and constraint satisfaction

using $\mathbf{k}$ to ensure the $\boldsymbol{\alpha}$ are properly normalized. It can be shown that, when the Gaussian kernel is used at the test input points to estimate the weight function, KLIEP converges to the optimal solution at a rate slightly slower than $O_p(n^{-1/2})$, assuming $n = m$.

KLIEP improves over CS and KMM in obtaining a general weight function that can be applied to points that are not available at training time. The method can be computationally intensive at times, especially when a large number of target samples are available. In this case, subsampling or clustering can be done to reduce computational cost in the estimation of the basis functions. KLIEP has similar issues to CS and KMM in that we still require $\mathcal{X} = \mathcal{Z}$, and further require F and G to be mutually absolutely continuous.

2.2.2 Dimension Reduction Methods

Much of the importance weighting methods have similar drawbacks, namely that they have issues handling cases in which source and target data not only come from different distributions, but perhaps even have differing underlying structures, indicating that $\mathcal{X} \neq \mathcal{Z}$ or $\text{supp}(\mathcal{Z})$ is not contained in $\text{supp}(\mathcal{X})$. In this case, transformations of the source or target data (or typically both) are desired to obtain a representation that is hopefully invariant to domain changes. Moreover, dimension reduction methods can potentially be used to further interpretation of results. Many dimension reduction methods require knowledge of the structure of the Grassmannian $\mathcal{G}(p, d)$, defined in Chapter 1.

The intermediate subspace approach (IS, [52]) seeks a latent feature representation by obtaining intermediate feature spaces that help to quantify the shift from the source to the target space. In IS, the latent variables are obtained by sampling points along a geodesic on the Grassmannian $\mathcal{G}$ between the d-dimensional subspace spanned by the source dataset and the d-dimensional subspace spanned by the target dataset. It can be written as follows: Using principal component analysis, estimate d-dimensional ($d < p = q$) representations of data $\mathbb{X}$ and $\mathbb{Z}$ as $\widetilde{\mathbb{X}} = \mathbb{X}\,\boldsymbol{\eta}_{\mathrm{x}}$ and $\widetilde{\mathbb{Z}} = \mathbb{Z}\,\boldsymbol{\eta}_{\mathrm{z}}$ for $\boldsymbol{\eta}_{\mathrm{x}}, \boldsymbol{\eta}_{\mathrm{z}} \in \mathcal{G}(p, d)$, and, using the geodesic along $\mathcal{G}(p, d)$ between $\boldsymbol{\eta}_{\mathrm{x}}$ and $\boldsymbol{\eta}_{\mathrm{z}}$, obtain intermediate transformations $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_K$ to use in finding representations

$$\widetilde{\mathbb{X}} \to \widehat{\mathbb{X}}_1 \to \cdots \to \widehat{\mathbb{X}}_{K+1} \text{ and } \widetilde{\mathbb{Z}} \to \widehat{\mathbb{Z}}_K \to \cdots \to \widehat{\mathbb{Z}}_0$$

where $\widehat{\mathbb{X}}_{K+1} = \mathbb{X}\,\boldsymbol{\eta}_{\mathrm{z}}$ and $\widehat{\mathbb{Z}}_0 = \mathbf{Z}\,\boldsymbol{\eta}_{\mathrm{x}}$. The newly acquired latent feature representation for the source data $\mathbb{X}$ is then just the concatenation of each of these matrices, which can be expressed as

$$\mathbb{X}^* = [\widetilde{\mathbb{X}} \ \cdots \ \widehat{\mathbb{X}}_{K+1}]$$

with a similar representation being acquired for data $\mathbb{Z}^*$. Partial least squares [53], hereafter called PLS, is performed to obtain a low-dimensional model operating on these expanded datasets. Some drawbacks to the IS method are its reliance on a large number of tuning parameters and the high dimensionality that must be overcome when many subspaces are desired.

Geodesic flow kernel (GFK, [54]) seeks improvements over the IS method. GFK

attempts to remove the need for sampling along the geodesic between the source and target subspaces and uses a kernel approach to mitigate the extreme dimensionality of IS when a large number of subspaces are used. As before, we have $\boldsymbol{\eta}_x, \boldsymbol{\eta}_z \in \mathbb{R}^{p\times d}$ as bases for the source and target subspaces, respectively, and write their orthogonal complements as $\boldsymbol{\eta}_x^{\perp}, \boldsymbol{\eta}_z^{\perp} \in \mathbb{R}^{p\times(p-d)}$. We recall the geodesic flow $\boldsymbol{\delta} : [0,1] \to \mathcal{G}(p,d)$ from Chapter 1 as

$$\boldsymbol{\delta}(t; \boldsymbol{\eta}_{\mathrm{x}}, \boldsymbol{\eta}_{\mathrm{z}}) = \boldsymbol{\eta}_{\mathrm{x}}\, \mathbf{U}_1\, \boldsymbol{\Gamma}(t) - \boldsymbol{\eta}_{\mathrm{x}}^{\perp}\, \mathbf{U}_2\, \boldsymbol{\Sigma}(t)$$

where $\mathbf{U}_1, \mathbf{U}_2, \boldsymbol{\Gamma}, \boldsymbol{\Sigma}$ are given by the generalized singular value decomposition

$$\boldsymbol{\eta}_{\mathrm{x}}^T\, \boldsymbol{\eta}_{\mathrm{z}} = \mathbf{U}_1\, \boldsymbol{\Gamma}\, \mathbf{V}^T,\ (\boldsymbol{\eta}_{\mathrm{x}}^{\perp})^T\, \boldsymbol{\eta}_{\mathrm{z}} = -\,\mathbf{U}_2\, \boldsymbol{\Sigma}\, \mathbf{V}^T\,.$$

Our goal is to use all $t \in (0,1)$ to obtain representations $\boldsymbol{\delta}(t)^T\, \mathbf{x}$ for $\mathbf{x}$ in the source domain. Computationally this is infeasible, so we proceed through a kernel approach where

$$< \Phi(\mathbf{u}), \Phi(\mathbf{v}) >= \int_0^1 (\boldsymbol{\delta}(t)^T\, \mathbf{u})^T (\boldsymbol{\delta}(t)^T\, \mathbf{v})\; dt = \mathbf{u}^T\, \mathbf{G}\, \mathbf{v}$$

with $\mathbf{G} \in \mathbb{R}^{p\times p}$ positive semidefinite and defined through matrices obtained using the previous generalized singular value decomposition. This kernel is used to perform prediction through using kernel nearest neighbor. Potential drawbacks to both IS and GFK are that the geodesic path between two subspaces may not be the most informative, especially when further labeling information is available. Moreover, all

of the issues with kernel methods that were outlined previously hold for GFK as well.

2.2.3 Empirical Comparison

We test the outlined methods against one another (as well as a baseline method) using synthetic datasets. The baseline method used is simply a least squares classifier using PCA found through both the given source and target data with no adaptation. Both unsupervised and semisupervised problems will be considered. All methods were outlined for the unsupervised case. In the semisupervised case, the data used for estimating the model is augmented with the given labeled target data; for all methods this is also included as the "target" data used to estimate the weights or transformation.

For CS, we learn density functions using a radial kernel density estimation procedure with a Gaussian kernel, i.e.,

$$\hat{f}_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{|\,\mathbf{H}\,|} K\left[\mathbf{H}^{-1}(\mathbf{x}_i - \mathbf{x})\right],$$

where K is a Gaussian kernel and $\mathbf{H}$ is a diagonal bandwidth matrix, with each diagonal element chosen according to the rule of thumb $\hat{h}_j = n^{-1/(p+4)}\hat{\sigma}_j$ with $\hat{\sigma}_j$ the estimated standard deviation for column j of $\mathbb{X}$ [55]. A similar density estimate is obtained for g. We choose a Gaussian kernel over a more efficient kernel (e.g., Epanechnikov) due to its infinite support.

For KMM, we take $\epsilon = 1 - n^{-1/2}$ and $W = 1000$ as in [50], but set $\sigma = 10^{-4}$

as it yielded better results in practice than the authors' suggestion of $\sigma = 10^{-1}$. For KLIEP, we take $\epsilon = 10^{-3}$ and $\sigma = 10^{-1}$. For IS, we set the number of intermediate subspaces to 8. These tuning parameters were found to give the best results. Other types of cross-validation can be performed for all methods, though for many of them this results in a large computational expense. To easily incorporate weights – and to reduce computational complexity – a one-vs-all least squares classifier [56] is estimated for all methods except GFK, which uses kernel nearest neighbor. For the baseline and instance-weighting methods, PCA is first performed to reduce the effect of the high dimensionality. For the case where $p \neq q$, features from the higher-dimensional space are transformed into the lower-dimensional space using PCA for all methods.

For the simulation studies, we generate 200 observations in $\mathbb{R}^6$ for the source data and 300 observations in $\mathbb{R}^4$ for the target data in three classes. We generate the source data as multivariate normal with zero mean and covariance matrix with

$$\mathbf{\Sigma}_{\text{ij}} = 0.5^{|i-j|} \text{ for } i, j = 1, \ldots, 6.$$

The target data is generated as a mixture of two normals, one with a mean vector of ones and covariance matrix

$$\mathbf{\Sigma}_{\text{ij}} = 0.5^{|i-j|} \text{ for } i, j = 1, \ldots, 4,$$

the other with a mean vector of negative ones and covariance matrix

$$\mathbf{\Sigma}_{\text{ij}} = 0.5^{2|i-j|} \text{ for } i, j = 1, \ldots, 4,$$

each with equal weight.

We generate both $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$ with random normal entries in $\mathbb{R}^{6\times 2}$ and $\mathbb{R}^{4\times 2}$, respectively, and take their orthogonalizations. The labels are generated as

$$y = \boldsymbol{\eta}_1^T \mathbf{x} - \boldsymbol{\eta}_2^T \mathbf{x} + \epsilon$$

where $\epsilon \sim N(0, .5^2)$. The response values y are then discretized into three categories by thresholding them at their one-third and two-thirds quantiles. Similar labels are generated for the target data using $\boldsymbol{\gamma}$.

Each study is run ten times, and the average recognition rate is recorded for all methods under consideration, along with the standard error. For the unsupervised case, we use half of the source data and half of the target data to build the model, while testing the model on the remaining half of both the source and target data. For the semisupervised studies, we build the model again on half of the source data, this time selecting three observations per class from the target data to use in parameter estimation. We then test this model on the remaining half of the source data and all 300 of the target data points. Recognition rates are given in Table 2.1 with standard errors given in parentheses.

Looking at the results on the target data, IS appears to perform best unsupervised settings when the dimension of the source and target data are the same, though it seems it is not significantly different from many of the competing methods.

Table 2.1: Recognition rates from simulation studies. Standard errors are given in parentheses. Maximum recognition rates given in bold.

Study	**Method**	$p = q$/**Source**	$p = q$/**Target**	$p \neq q$/**Source**	$p \neq q$/**Target**
No labeled target data	Baseline	**54.73** (0.95)	36.16 (2.36)	54.30 (0.89)	36.23 (2.64)
	CS	54.14 (0.96)	36.59 (2.38)	54.14 (0.90)	35.92 (2.60)
	KMM	54.60 (0.98)	36.57 (2.33)	**54.34** (0.90)	**36.38** (2.66)
	KLIEP	54.68 (0.96)	36.25 (2.35)	54.30 (0.89)	36.22 (2.65)
	IS	54.16 (1.03)	**37.23** (2.29)	53.70 (0.99)	36.36 (2.32)
	GFK	49.43 (1.14)	34.95 (1.74)	52.63 (0.95)	34.77 (1.67)
3 observations per class from target data	Baseline	**54.06** (0.98)	41.36 (2.46)	**54.80** (0.94)	33.89 (2.42)
	CS	51.35 (1.10)	**46.52** (2.40)	52.18 (1.08)	39.16 (2.34)
	KMM	52.09 (1.14)	44.44 (2.42)	53.89 (0.97)	35.21 (2.44)
	KLIEP	53.92 (0.99)	41.44 (2.45)	54.67 (0.93)	33.97 (2.42)
	IS	52.98 (1.07)	40.29 (2.36)	54.34 (1.02)	**40.07** (2.40)
	GFK	51.17 (1.19)	35.83 (1.75)	52.57 (1.08)	35.37 (1.70)

In the semisupervised case, IS and CS seem to perform competitively over the alternatives. When the dimension between the source and target differ, CS and KMM – both instance-weighting methods – perform best in the unsupervised case while CS and IS perform best in the semisupervised case. While no method appears to be a clear winner in this simulation, if some labeled information is available from the target space, potential improvements over a baseline method are possible.

2.2.4 Discussion

Many methods are available to estimate model parameters given a regression function with high-dimensional, highly-correlated inputs. The EDE performs well in a classification task on a dataset that included images as predictors. All of the approaches consider examples in which the underlying structure of the predictors is assumed to exist but is not known explicitly. We will see a method for incorporating known structure in the following chapter.

When the distribution of features is heterogeneous across training and testing,

modifications can be made to aid in prediction. We have two different paradigms: instance weighting methods, in which each observation is assigned a weight to even out distributional differences; and dimension reduction methods, in which feature spaces are sought that minimize the discrepancy between training and testing sets. IS performs well when source and target dimensions are the same, while instance weighting methods tend to benefit from knowing some response values from the target. In any case, using some method to account for a domain shift can be a first step to improving predictive performance in pattern recognition tasks.

3

Regression on the Grassmannian

3.1 Introduction

The exterior derivative estimator outlined in the previous chapter was able to perform well when dealing with high-dimensional data that took on a low-dimensional structure, though the underlying structure was not assumed to be known a priori. In the following chapter, we outline a method that considers a specific manifold structure that arises often in computer vision problems and test its performance against various alternatives.

3.2 Methodology

As in Chapter 2, we assume our goal is to estimate a regression function m given predictors lying on a manifold. In all examples, we will focus on problems in the computer vision literature. In this problem setting, we see the manifold assumption can be crucial, as visual information has a rich underlying structure. As seen in Chapter 1, depending on the problem, this structure can either be taken as known or unknown. Cases in which we do have this prior information include analysis using landmark points or dynamic models in classification tasks using video data, with the structure coming from a preprocessing of the data.

Algorithm 1 Calculation of the projection matrix in the regularization for the EDE [9]. When used in conjunction with an ℓ_2 penalty, this matrix will penalize regression coefficients for not lying parallel to the tangent space formed by $\mathbf{x}_1, \ldots, \mathbf{x}_n$.

1: **Estimate:** d with $\hat{d}$ using maximum likelihood [44]
2: $\widehat{\mathbf{C}} \leftarrow \sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T / n$
3: $\widehat{\mathbf{C}} \leftarrow [\widehat{\mathbf{R}}\ \widehat{\mathbf{N}}] \cdot \widehat{\mathbf{\Lambda}} \cdot [\widehat{\mathbf{R}}\ \widehat{\mathbf{N}}]^T$ eigenvalue decomposition of $\widehat{\mathbf{C}}$ with $\widehat{\mathbf{R}} \in \mathbb{R}^{p \times \hat{d}}$, $\widehat{\mathbf{N}} \in \mathbb{R}^{p \times (p-\hat{d})}$, $\widehat{\mathbf{\Lambda}}$ a diagonal matrix
4: $\hat{\mathbf{\Pi}} \leftarrow \widehat{\mathbf{N}}\widehat{\mathbf{N}}^T$
5: $\widehat{\mathbf{P}} \leftarrow \mathrm{diag}(0, \hat{\mathbf{\Pi}})$

3.2.1 EDE with Prior Structure

As described in Chapter 2, the EDE is a useful approach to estimating a regression function when the predictors are thought to lie on a manifold, though in the previous chapter the structure was assumed unknown so that the manifold had to be estimated. The exterior derivative was estimated by locally penalizing the regression coefficients for not falling onto the d largest principal components (cf. PCR in which predictors are projected directly onto these components). We summarize the estimation of the projection matrix $\widehat{\mathbf{P}}$ for a globally linear manifold in Algorithm 1.

As the EDE method given in [9] is largely concerned with the case where predictors lie on an *unknown* manifold, we modify the approach to take advantage of a priori information regarding the structure of the predictors, such as in the examples of Chapter 1. We extend the EDE method to cases in which prior knowledge is available, with only the regularization needing modification. In constructing the EDE, the projection orthogonal to the tangent space is estimated with the data due to our not knowing the structure in the predictors, whereas here we seek a direct

projection of the coefficient vectors. To find this projection, we first look at the structure of $\mathcal{G}(r,s)$. Recall $\mathcal{G}(r,s)$ as the set of all s-dimensional subspaces of $\mathbb{R}^r$, i.e., the quotient space

$$\mathcal{G}(r,s) = \mathcal{R}(r,s)/\sim$$

where $\mathcal{R}(r,s)$ is the space of all $r \times s$ matrices of rank s, and, for $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{r\times s}$, $\mathbf{U} \sim \mathbf{V}$ if there exists a nonsingular $\mathbf{L} \in \mathbb{R}^{s\times s}$ such that $\mathbf{V} = \mathbf{U}\,\mathbf{L}$ [2]. The tangent structure of $\mathcal{G}(r,s)$ is slightly different from that of a manifold formed by data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ due to this quotient space representation. Rather than tangent spaces to points on $\mathcal{G}(r,s)$, we seek tangent spaces to equivalence classes of points, which for $\mathcal{G}(r,s)$ can be identified with semi-orthogonal matrices $\mathbf{U} \in \mathbb{R}^{r\times s}$. The tangent space to the equivalence class of a point is known as the *vertical space*, and its orthogonal complement is called the *horizontal space* [57]. For two orthogonal matrices $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{r\times s}$ representing points in $\mathcal{G}(r,s)$, projection of a matrix $\mathbf{U}$ into the horizontal space at a point $\mathbf{V}$ can be done with the operator

$$\pi_v(\mathbf{U}) = \left(\mathbf{I}_\mathrm{r} - \mathbf{V}\,\mathbf{V}^T\right)\mathbf{U}$$

where $\mathbf{I}_\mathrm{r}$ is the $r\times r$ identity matrix. In this case, if the predictors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{G}(r,s)$, we think of the regression coefficients $\boldsymbol{\beta}_1^M$ as lying in $\mathbb{R}^{r\times s}$ to allow for a projection of $\boldsymbol{\beta}_1^M$ into the horizontal space using π_v. In order to perform the regression, we reshape predictors $\mathbf{x}$ and coefficients $\boldsymbol{\beta}$ by concatenating column-wise so that $\mathbf{x}, \boldsymbol{\beta} \in \mathbb{R}^{rs}$. The estimate of the projection matrix $\widehat{\mathbf{P}}$ for the regularization in this case is given

Algorithm 2 Calculation of the projection matrix for EDE with Grassmann prior. This regularization assumes the data $\mathbf{x}_1, \ldots, \mathbf{x}_n$ lie on a Grassmannian $\mathcal{G}(r, s)$ and will penalize a projection of the regression coefficients into the horizontal space of $\mathcal{G}(r, s)$.

1: **Compute:** orthogonalization $\bar{\mathbf{x}}_*$ of $\bar{\mathbf{x}} \in \mathbb{R}^{r \times s}$ using singular value decomposition
2: $\hat{\mathbf{\Pi}}^M \leftarrow (\mathbf{I}_r - \bar{\mathbf{x}}_* \bar{\mathbf{x}}_*^T)$
3: $\hat{\mathbf{\Pi}} \leftarrow \text{diag}(\hat{\mathbf{\Pi}}^M, \ldots, \hat{\mathbf{\Pi}}^M)$, a block-diagonal matrix such that $\hat{\mathbf{\Pi}} \in \mathbb{R}^{rs \times rs}$
4: $\hat{\mathbf{P}} \leftarrow \text{diag}(0, \hat{\mathbf{\Pi}})$

in Algorithm 2.

3.2.1.1 Bayesian Interpretation

For the EDE with Grassmannian data, estimates for β can be found by computing

$$\arg\min_{\boldsymbol{\beta}} \; || \, \mathbf{y} - \mathbb{X} \, \boldsymbol{\beta} \, ||_2^2 + \lambda \cdot ||(\mathbf{I}_{\text{n}} - \bar{\mathbf{x}}_* \bar{\mathbf{x}}_*^T) \, \boldsymbol{\beta}^{\text{M}} \, ||_F^2 \tag{3.1}$$

where as before $\bar{\mathbf{x}}_*$ is the orthogonalized sample mean of the predictors, $\boldsymbol{\beta}^{\text{M}} \in \mathbb{R}^{r \times s}$ is the "matrix" version of $\boldsymbol{\beta}_1$, and $|| \cdot ||_F^2$ is the squared Frobenius norm defined as

$$|| \, \mathbf{U} \, ||_F^2 = \text{tr}(\mathbf{U} \, \mathbf{U}^T).$$

This penalization term can be interpreted as placing a "Procrustean" prior on the parameters $\boldsymbol{\beta}^{\text{M}}$. In other words, the estimate for β^M obtained by optimizing (3.1) above can be obtained as the Bayes posterior mode under the prior

$$f(\boldsymbol{\beta}^{\text{M}}; \lambda) = c \cdot \exp\{-\lambda \cdot g\left(\bar{\mathbf{x}}_*, \boldsymbol{\beta}^{\text{M}}\right)\}$$

where $g(\mathbf{U}, \mathbf{V}) = \text{tr}\left(\mathbf{V}^T\mathbf{V} - \mathbf{U}^T\mathbf{V}\mathbf{V}^T\mathbf{U}\right)$ and c is a normalizing constant. This is called a "Procrustean" prior due to the fact that g above is similar to the Procrustes distance metric $g_P(\mathbf{U}, \mathbf{V}) = \text{tr}\left(\mathbf{I}_s - \mathbf{U}^T\mathbf{V}\mathbf{V}^T\mathbf{U}\right)$ given in [2]; in fact, it will hold locally that $(\boldsymbol{\beta}^{\text{M}})^T(\boldsymbol{\beta}^{\text{M}}) \approx \mathbf{I}_s$ since $\boldsymbol{\beta}^{\text{M}}$ should lie on the tangent space to the manifold on which $\mathbf{x}_1, \ldots, \mathbf{x}_n$ reside [i.e., $\mathcal{G}(r, s)$], and any point $\mathbf{x}_i$ on this manifold satisfies $\mathbf{x}_i^T\mathbf{x}_i = \mathbf{I}_s$.

3.2.2 The Fréchet Mean

The Fréchet sample mean is a useful concept for defining the concept of the sample mean on a manifold [58]. For an i.i.d. sample $\mathbf{x}_1, \ldots, \mathbf{x}_n$ on a metric space $(\mathcal{M}, \delta)$, we define the Fréchet sample mean set as the set of all minimizers in $\mathcal{M}$ of the function

$$Q(\mathbf{p}) = \frac{1}{n}\sum_{i=1}^{n}\delta^2(\mathbf{x}_{\text{i}}, \mathbf{p}).$$

If there exists only one minimizer to this function, we call this the Fréchet mean.

The general procedure for computing a mean of a set of values on a Riemannian manifold is to use an iterative procedure: each point is projected into the tangent space about a candidate mean value, the sample mean in this tangent space is computed, and then this sample mean is projected back some distance along the geodesic between it and the previous candidate mean value. This procedure [59] is outlined in Algorithm 3. It uses the notions of geodesic flow and exponential maps defined in Chapter 1.

Algorithm 3 Iterative computation of the Fréchet mean (also called Karcher mean) of a set of points [59].

given $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^{r \times s}$ as elements of $\mathcal{G}(r, s)$
initialize $\boldsymbol{\mu}_0 = \mathbf{x}_1$, $\eta = .5$, $\tau \in (0, 1)$, $j = 0$, and $t = 1$
while $t > \tau$ **do**
  for $i = 1, \ldots, n$ **do**
    $\boldsymbol{\nu}_i \leftarrow \exp^{-1}(\mathbf{x}_i; \boldsymbol{\mu}_j)$
  end for
  $\bar{\boldsymbol{\nu}} \leftarrow \sum_i \boldsymbol{\nu}_i / n$
  $\boldsymbol{\mu}_{j+1} \leftarrow \exp(\eta \cdot \bar{\boldsymbol{\nu}}; \boldsymbol{\mu}_j)$
  $t \leftarrow || \boldsymbol{\mu}_j - \boldsymbol{\mu}_{j+1} ||$
  $j \leftarrow j + 1$
end while
return $\bar{\mathbf{x}}_1 = \boldsymbol{\mu}_j$

3.2.3 Parameter Selection

3.2.3.1 Regularization

In generalized cross-validation [60], the regularization parameter λ is chosen as the minimum of the objective

$$V(\lambda) = \frac{1}{n} ||(\mathbf{I}_\mathrm{n} - \mathbf{A}(\lambda))\, \mathbf{y}\,||^2 \Big/ \left[\frac{1}{n}\mathrm{tr}(\mathbf{I}_\mathrm{n} - \mathbf{A}(\lambda))\right]^2$$

and

$$\mathbf{A}(\lambda) = \mathbb{X}(\mathbb{X}^T\,\mathbb{X} + \lambda \cdot \mathbf{I}_\mathrm{p})^{-1}\,\mathbb{X}^T\,.$$

An efficient algorithm for selecting λ is based on the singular value decomposition of $\mathbb{X}$ as $\mathbb{X} = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^T$. The quantity $V(\lambda)$ can then be rewritten as

$$V(\lambda) = n \sum_{i=1}^{n} \left(\frac{\lambda}{\nu_i + \lambda} \right)^2 z_i^2 \Bigg/ \left[n - p + \sum_{i=1}^{n} \frac{\lambda}{\nu_i + \lambda} \right]^2$$

where $(z_1, \ldots, z_n)^T = \mathbf{U}^T \mathbf{y}$ and ν_i for $i = 1, \ldots, n$ are the eigenvalues of $\mathbb{X}\mathbb{X}^T$. This allows for ease of computation of the objective to facilitate a global search for the optimal λ. In experiments we restrict our search within the set $\{10^{-3}c : c = 1, 2, \ldots, 10^6\}$.

In our case, $\mathbf{P}$ is idempotent and, by virtue of its construction, will be for all practical purposes invertible. We apply the same parameter selection as above, except we write

$$\begin{aligned} \widetilde{\mathbf{A}}(\lambda) &= \mathbb{X}(\mathbb{X}^T \mathbb{X} + \lambda \cdot \mathbf{P})^{-1} \mathbb{X}^T \\ &= \mathbb{X}(\mathbb{X}^T \mathbb{X} + \lambda \cdot \mathbf{P}^2)^{-1} \mathbb{X}^T = \widetilde{\mathbb{X}}(\tilde{\mathbb{X}}^T \widetilde{\mathbb{X}} + \lambda \cdot \mathbf{I}_{\mathrm{p}})^{-1} \tilde{\mathbb{X}}^T \end{aligned}$$

where $\widetilde{\mathbb{X}} = \mathbb{X}\,\mathbf{P}^{-1}$. This will allow for a more efficient parameter selection. For multi-class classification problems, we choose regularization parameters for each of C labels using $y^k = \mathbf{1}(y = k)$ for $k = 1, \ldots, C$.

3.2.3.2 Localization

In the case of predictors on a Grassmannian, performing local regression at the point $\mathbf{x}_0$ will require the computation of weights. Using the definition of the kernel density estimate from [2] as a baseline, we will use weights

$$w(\mathbf{x}_\mathrm{i}, \mathbf{x}_0) = \exp\{-\mathrm{tr}[\mathbf{B}^{-1}(\mathbf{I}_\mathrm{s} - \mathbf{x}_\mathrm{i}^T \mathbf{x}_0 \mathbf{x}_0^T \mathbf{x}_\mathrm{i})]\}$$

with $\mathbf{B}$ an $s \times s$ bandwidth matrix. In [2], the bias and variance for the estimated density at $\mathbf{x}_0$ are $O(\mathbf{B})$ and $O([n|\,\mathbf{B}\,|^{(r-s)/2}]^{-1})$, respectively. To achieve a tradeoff between this bias and variance, we pick $\mathbf{B}$ to be diagonal and on the order of $\alpha = n^{-1/[s(r-s)/2+2]}$. In experiments, the optimal bandwidth will be chosen among the set $\{10^{-\delta}\alpha : \delta = 0, \ldots, 5\}$ to minimize the generalized cross-validation criterion [61]

$$V(\mathbf{B}) = \frac{n||(\mathbf{I}_\mathrm{n} - \mathbf{H})\,\mathbf{y}\,||^2}{[n - \mathrm{tr}(\mathbf{H})]^2}.$$

where $\mathbf{H}$ is the matrix such that $(\hat{y}_1, \ldots, \hat{y}_n)^T = \mathbf{H}\,\mathbf{y}$. We take only a small number of values in the candidate set because of the computational overhead in computing $\mathbf{H}$ for each potential bandwidth.

3.3 Case Studies

We use two datasets to assess the performance of the proposed method: the FG-NET dataset [3], and the video sequence dataset used in [23], both described in Chapter 1. The FG-NET database consists of images of 82 separate individuals' faces, with a total of 1002 images in the database, 571 of which correspond to males and 431 females. The video sequence dataset consists of videos of 16 subjects' faces taken at two different times (the "gallery" and the "probe" sequences). For

convenience, we truncate each video at 20 frames, with individuals having either 4, 8, or 16 different 20-frame videos.

3.3.1 Localization on $\mathcal{G}(r, s)$

Due to the inherent structure in our predictors the method for obtaining weights for local regression is modified to take into account points lying on a Grassmannian. In [2], the Procrustes distance metric $g_P(\mathbf{U}, \mathbf{V}) = \text{tr}(\mathbf{I}_s - \mathbf{U}^T \mathbf{V} \mathbf{V}^T \mathbf{U})$ is given for $\mathbf{U}, \mathbf{V} \in \mathcal{G}(r, s)$. We use this distance metric as an argument to a Gaussian kernel to weight observations when localizing regression. Typically a modified version of this distance – $g_P^*(\mathbf{U}, \mathbf{V}) = [g_P(\mathbf{U}, \mathbf{V}) + g_P(\mathbf{V}, \mathbf{U})]/2$ – is used due to the fact that g_P is not symmetric in its arguments. We use this distance metric in a nearest neighbor classifier for comparison. Choice of the Gaussian kernel was made due to ease of its computation.

3.3.2 Related Methods

If the manifold structure is known a priori as in the case of $\mathcal{G}(r, s)$, we can perform regression using classical least squares by first projecting all observations to the tangent space about a point on the manifold (e.g., $\bar{\mathbf{x}}_*$). Since the tangent space to a point on $\mathcal{G}(r, s)$ has the structure of Euclidean space, no assumptions on the structure of the predictors are violated and the well-known least squares solution for $\boldsymbol{\beta}$ can be computed as [62]

$$\hat{\boldsymbol{\beta}} = [\mathbb{X}(\boldsymbol{\mu})^T \mathbb{X}(\boldsymbol{\mu})]^{-1} \mathbb{X}(\boldsymbol{\mu})^T \mathbf{y}$$

where $\mathbb{X}(\boldsymbol{\mu})$ are the observations $\mathbb{X}$ transformed via inverse exponential map to lie on the tangent space at the point $\boldsymbol{\mu}$. The algorithm for computing this inverse exponential map is given in [59] and depends on a single parameter t. In our experiments, $\boldsymbol{\mu}$ is taken to either be the orthogonalized sample mean or the local Fréchet mean, depending on whether global linearity is assumed, and t is taken to be one.

Procrustes nearest-neighbor was also considered as another method that takes the explicit, known manifold structure into account. This method finds the observation in the training dataset that minimizes the Procrustes distance [$g_P^*(\mathbf{U}, \mathbf{V})$ given above] between it and the given test point. The label of the point in the training set is then used as the estimated label for the given test point.

3.3.3 Linearity Assumption on FG-NET

The Grassmannian $\mathcal{G}(r, s)$ is a nonlinear manifold; however, a useful property of manifolds is that locally they behave like Euclidean space. For the example of age estimation, the Grassmannian structure comes from predictors as landmark points on a face. Thus it can be assumed that they do not have a high variability: an individual's eyes will typically appear above the nose and mouth and not be spaced arbitrarily far apart or close together.

To test the linearity assumption, 1000 pairs of points were chosen with replacement at random from the dataset and the quantity $\mathbf{x}_i^T \mathbf{x}_j$ was computed. The

Table 3.1: Comparison between means and standard deviations of $\mathbf{x}_i^T \mathbf{x}_j$ for FG-NET dataset for i, j randomly chosen from $\{1, \ldots, n\}$ (left) and randomly generated observations (right). Note for $\mathbf{x}_i \in \mathcal{G}(r, s)$ we have $\mathbf{x}_i^T \mathbf{x}_i = \mathbf{I}_s$.

FG-NET dataset		**Random uniform data**	
.9987 (.0012)	-.0010 (.0321)	.0009 (.0037)	.0001 (.0035)
.0031 (.0320)	.9661 (.0396)	-.0000 (.0035)	.0009 (.0033)

Table 3.2: Method for generating a single uniform random variate $\mathbf{Y}_i$ on $\mathcal{G}(r, s)$ [2].

- Generate rs random standard normal variates $u_1, \ldots, u_{rs} \sim N(0, 1)$;
- Form random variates into matrix $\mathbf{U} = [\mathbf{u}_1 \ \ldots \ \mathbf{u}_s]$ where $\mathbf{u}_1 = [u_1, \ldots, u_r]^T$;
- Compute matrix $\mathbf{Z} = \mathbf{U}(\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T$;
- Form $\mathbf{Y}_i = [\mathbf{z}_1 \ \ldots \ \mathbf{z}_s]$ where $\mathbf{z}_i$ are the columns of $\mathbf{Z}$.

means and standard deviations of each element of this matrix are given in Table 3.1. For comparison, the same was done with data generated uniformly at random on $\mathcal{G}(r, s)$, with the method for obtaining these random observations $\mathbf{Y}_i$ outlined in Table 3.2. This linearity assumption is used to increase computational efficiency, though localization can potentially yield better results.

Since observations corresponding to normalized landmark points are contained within a small (read: approximately Euclidean) subset of the Grassmannian, a simpler computation of an approximation to the Fréchet mean can be done as given in Algorithm 4. Instead of using an iterative procedure that relies on projecting and reprojecting sample points (using the inverse exponential and exponential map, respectively), the sample mean of the data can be taken and then orthogonalized to ensure it lies on the Grassmannian. This greatly improves computation time and additionally requires fewer tuning parameters than computation of the Fréchet mean

Algorithm 4 Computation of the orthogonalized sample mean.

given $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{G}(r, s) \subset \mathbb{R}^{r \times s}$
compute $\bar{\mathbf{x}} \leftarrow \sum_i \mathbf{x}_i / n$
let $\mathbf{v}_k$ be such that $\bar{\mathbf{x}}\mathbf{v}_k = \lambda_k \mathbf{u}_k$ and $\bar{\mathbf{x}}^* \mathbf{u}_k = \lambda_k \mathbf{v}_k$ with $\lambda_1 \geq \ldots \geq \lambda_r$
return $\bar{\mathbf{x}}_2 = [\mathbf{u}_1 \ \ldots \ \mathbf{u}_s]$

using Algorithm 3.

An empirical comparison between these two methods was performed using the FG-NET database [3], the results given in Table 3.3. A random selection of 30, 100, and 1000 observations were chosen from the FG-NET database, and Algorithm 3 was performed using 2, 4, 6, and 8 iterations. Meanwhile, the sample mean was also computed using Algorithm 4. The Frobenius norm between the two computed means was calculated, along with the computation times of both algorithms. It is interesting to note that, as the number of iterations increases, Algorithm 3 approaches the value obtained by simply orthogonalizing the sample mean, and as these iterations increased, the gap between computation times widened. On 1000 points using 8 iterations, it takes over three seconds to compute the mean using Algorithm 3, compared with .002 seconds using the alternative method. Fig. 3.1 shows a graphical comparison between the landmark points of the sample mean of the entire dataset computed using Algorithm 3 with 10 iterations and the proposed, simpler method, showing these methods obtain similar configurations.

3.3.4 Experimental Setup

The alternative methods used for comparison for age estimation where the structure of the predictors is unknown were ordinary least squares (OLS), ridge re-

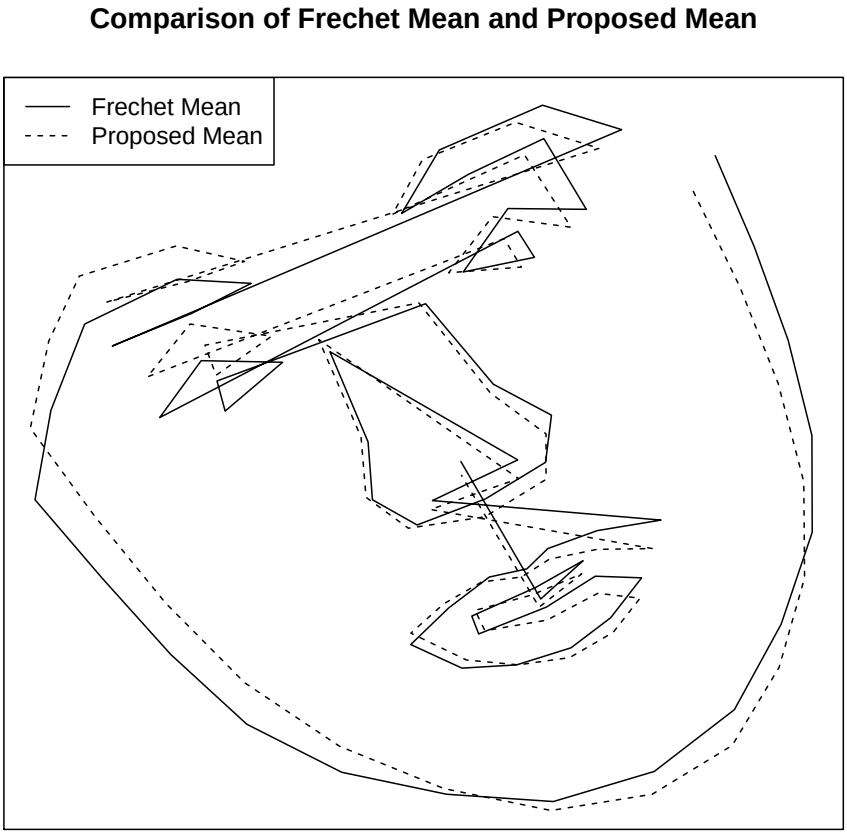


Fig. 3.1: Comparison of mean face obtained via Algorithm 3 (Fréchet mean with 10 iterations) and the proposed mean.

Table 3.3: Comparison between Algorithm 3 and the orthogonalized sample mean. For $n = 30, 100, 1000$ samples with replacement from the dataset, both Algorithm 3 and the proposed mean were computed, and the Frobenius norm between estimated means as well as MSE between computation times are reported.

		2 iterations	**4 iterations**	**6 iterations**	**8 iterations**
Error between Algorithm 3 and proposed					
$n = 30$	Error	.012 (.006)	.011 (.005)	.009 (.004)	.008 (.004)
$n = 100$	Error	.012 (.006)	.010 (.005)	.009 (.005)	.007 (.004)
$n = 1000$	Error	.012 (.006)	.011 (.005)	.008 (.004)	.007 (.003)
Computation times					
$n = 30$	Algorithm 3	.035 (.000)	.070 (.001)	.105 (.001)	.139 (.002)
	Proposed	**.000** (.000)	**.000** (.000)	**.000** (.000)	**.000** (.000)
$n = 100$	Algorithm 3	.096 (.009)	.187 (.002)	.278 (.002)	.369 (.002)
	Proposed	**.000** (.000)	**.000** (.000)	**.000** (.000)	**.000** (.000)
$n = 1000$	Algorithm 3	.837 (.009)	1.67 (.011)	2.49 (.003)	3.32 (.010)
	Proposed	**.003** (.001)	**.003** (.001)	**.003** (.001)	**.002** (.001)

gression (RR), principal components regression (PCR) [28] and partial least squares (PLS); these methods were compared to the exterior derivative estimator (EDE). In the case where landmark points are used as predictors, regression performed on points projected to the tangent space about the estimated mean (REM) as described in Section 3.3.2 is also used for comparison, and the exterior derivative estimator with prior (EDEwP) method is used as a proposed method. Regression on points embedded using locality preserving projections [42] has been used on this problem, but in this case was shown not to yield competitive results. All forms of regression were performed on either the predictors or the embedded predictors. For age estimation, improvements in prediction can be gained by additionally including the square of each predictor in the model [22], but this was not considered in this analysis.

3.3.5 Age Estimation

For age estimation on the FG-NET dataset, feature extraction was used to obtain predictors whose structure is not explicitly known in advance. To obtain features, each image was converted to normalized grayscale taking values between 0 and 1, and the Viola-Jones face detection algorithm [40] was used to discard much of the noise and unwanted information contained in the background. Finally, a histogram of oriented gradients (HOG, [63]) feature extraction method with 9 bins on 8×8 patches was used to generate predictors $\mathbf{x}_1, \ldots, \mathbf{x}_{1002} \in \mathbb{R}^{576}$. By construction HOG features have unit norm, but we standardize the data obtained so that the predictors lie on an "unknown" manifold. See Chapter 1 for details. Here each observation is labeled with ages y ranging from 0 to 69. In various experiments, performing regression on $\sqrt{y}$ yielded more accurate predictions; using this as a response variable has the added benefit that predictions of an individual's age will always be nonnegative.

A popular objective in the age estimation literature for assessing algorithm performance is to use a hold-one-person-out cross-validation and report the mean absolute error (MAE). In other words, 82 separate trials are performed where for each trial, the test dataset consists of all images of one specific individual while the training dataset is composed of the remaining 81 individuals. This method of assessment, hereafter referred to as Framework 3, gives a good indication as to how well methods are performing, but as an objective for both parameter tuning and performance assessment it can be prone to overfitting. This cross-validation framework is closer in spirit to a jackknife cross-validation, and obtaining a randomized

split between training and testing data may give a better idea of how the methods are performing relative to one another, as well as deter overfitting. We propose two alternative frameworks: Framework 1 chooses 5 test points at random for testing and uses the remaining observations for training the model; Framework 2, to be more consistent with hold-one-person-out cross-validation, does the same as Framework 1 but instead of training on the remaining individuals, each observation corresponding to a person in the testing set is removed from the training set and models are then built on this modified dataset. In both age estimation studies, a local model is learned for comparison using the hold-one-out cross-validation with the bandwidth selection described in Section 3.2.3.

We use both Framework 1 and Framework 2 100 times and report the average and standard error of the MAE for each method in Tables 3.4 and 3.5. In the case in which the structure of the predictors is unknown, ridge regression outperforms the alternatives, with the EDE a close second. Using the landmark data gives an overall improvement in performance for all methods. In this case, the EDEwP outperforms all alternatives with ridge regression a close second and the EDE not far behind. In both cases, the local regression yields only slightly better results than assuming global linearity.

3.3.6 Classification on FG-NET

For age and gender classification, global linearity is assumed. The improvements in performance found in age estimation due to the localization were not

Table 3.4: Age estimation results for various testing frameworks performed on HOG data in which the structure is unknown. Minimum mean absolute errors (MAEs) are given in bold.

Model	Framework 1 MAE(SE)	Framework 2 MAE(SE)	Framework 3 MAE	Local MAE
OLS	10.75 (0.45)	11.28 (0.47)	10.84	10.80
RR	**7.59** (0.31)	**7.89** (0.32)	**8.22**	**7.93**
PCR	7.90 (0.31)	8.07 (0.32)	8.38	8.66
PLS	8.46 (0.34)	8.76 (0.35)	8.88	9.04
EDE	7.68 (0.31)	7.97 (0.32)	8.29	8.12

Table 3.5: Age estimation results for various testing frameworks performed on landmark data in which the structure is known. Minimum mean absolute errors (MAEs) are given in bold.

Model	Framework 1 MAE(SE)	Framework 2 MAE(SE)	Framework 3 MAE	Local MAE
OLS	6.37 (0.31)	7.00 (0.34)	6.46	6.45
RR	6.06 (0.31)	6.50 (0.33)	6.11	6.10
PCR	7.33 (0.35)	7.54 (0.35)	6.87	6.95
PLS	6.20 (0.31)	6.57 (0.33)	6.15	6.12
REM	10.73 (0.48)	10.96 (0.49)	9.65	8.74
EDE	6.07 (0.31)	6.51 (0.33)	6.12	6.13
EDEwP	**6.03** (0.31)	**6.46** (0.32)	**6.11**	**6.09**

enough to warrant the computational burden. The methods used for comparison in this case are those that were used in age estimation, as well as a Procrustes nearest neighbor classifier (PRO) as described in Section 3.3.2. For the methods PCR, PLS, and EDE, an estimate of the dimension of the predictors is obtained using the maximum likelihood method [44].

Two classification experiments were performed on the FG-NET dataset: age group classification and gender classification. For age group classification, the observations are placed into three separate categories corresponding to age $0-8$, age

Table 3.6: Gender classification results for different testing frameworks performed on landmark data in which the structure is known. Maximum recognition rates are given in bold.

Model	Framework 1 Rec. Rate (SE)	Framework 2 Rec. Rate (SE)	Framework 3 Rec. Rate
OLS	**75.60** (2.06)	65.00 (2.08)	63.87
RR	75.20 (1.84)	67.60 (2.05)	64.57
PCR	61.60 (1.83)	58.60 (1.89)	57.78
PRO	60.80 (2.22)	60.80 (2.22)	56.99
PLS	**75.60** (1.79)	66.00 (1.92)	63.17
REM	60.80 (2.36)	56.40 (2.40)	54.19
EDE	75.20 (1.84)	**67.80** (2.03)	**64.67**
EDEwP	74.80 (1.83)	**67.80** (2.01)	64.57

$9 - 18$, and age 19 and up. The testing framework is the same as in age estimation, but instead of reporting the mean absolute errors, the proportions of correctly classified values will be given along with the corresponding standard errors (where applicable). For ease of illustration, these classification experiments were performed only on the landmark data.

In Tables 3.6 and 3.7 we see that although the proposed method does not perform universally best in gender classification, it is still comparable to the best methods and performs best in the second framework along with the EDE. In the third framework, it performs second best, along with ridge regression. For age group classification, results for the proposed method are more promising with it performing best in all frameworks.

3.3.7 Video-Based Face Recognition

We obtain feature vectors for the video-based face recognition dataset by first resizing images to 9×8 matrices of grayscale values between 0 and 1. If we were

Table 3.7: Age group classification results for various testing frameworks performed on landmark data in which the structure is known. Maximum recognition rates are given in bold.

Model	Framework 1 Rec. Rate(SE)	Framework 2 Rec. Rate(SE)	Framework 3 Rec. Rate
OLS	67.60 (2.18)	65.60 (2.20)	65.47
RR	71.40 (2.13)	70.80 (2.15)	68.76
PCR	65.20 (2.04)	65.00 (1.96)	63.47
PRO	34.20 (2.26)	34.20 (2.26)	31.44
PLS	70.00 (2.28)	68.20 (2.38)	67.96
REM	41.60 (2.02)	42.40 (2.06)	41.72
EDE	71.40 (2.13)	70.80 (2.15)	68.56
EDEwP	**72.00** (2.13)	**71.80** (2.11)	**68.96**

to concatenate the rows of each frame, and then concatenate all 20 frames, we would obtain features in $\mathbb{R}^{1440}$. Using the method described in Section 1.1.4.1 of Chapter 1, we are able to reduce our dimensionality to $\mathbb{R}^{144}$ while incorporating the Grassmannian structure into the predictors.

Because multiple observations are taken of the same subject, two methods are used to sidestep the issue of nonindependence of observations. First, an experimental setup is used in which the training data consists of an observation selected at random for each of the individuals from the gallery data, and 16 observations are chosen from the probe data at random for testing. This is done using the probe data for training and the gallery data for testing as well, and each test is replicated 100 times with the proportion of correctly classified values given along with the standard errors. The second experimental setup uses the Fréchet mean of the observations for each individual from the gallery as points in the training data, again tested on 16 observations chosen at random from the probe set; this is repeated using the probe as training data. Finally, the third framework corresponds to simply using the entire

gallery as training data to test on the probe (and vice versa) and is presented for comparison purposes.

As described in [9], cases in which $p \gg n$ present difficulty when estimating the EDE. For this dataset, we take $n = 16$ and $p = 144$, so to remedy this, covariance thresholding is performed where all elements of the covariance matrix that are less than the 0.75 quantile of the absolute value of the elements are set to zero. Both the regular EDE and the thresholded EDE (EDET) are reported.

Table 3.8 gives results on this dataset. When using the gallery as training data, in the first framework both ridge regression and partial least squares perform best, with the proposed method close behind. The proposed method performs best in the other two frameworks. When using the probe as training data, the best methods perform much better than using the gallery as training data. Ridge regression, partial least squares, and the EDEwP seem to be the only worthwhile methods in the first two frameworks, with the EDEwP performing best. When using the whole probe as training data, ridge regression, partial least squares, and the EDE all perform best, with the proposed method fewer than three percentage points away. It is interesting to note that the thresholding of the covariance matrix actually yields much worse results in classification results for this problem, possibly due to its discarding of a lot of information.

Table 3.8: Video-based face recognition results for various testing frameworks performed on appearance data in which the structure is known. Maximum recognition rates are given in bold.

Model	Framework 1 Rec. Rate (SE)	Framework 2 Rec. Rate (SE)	Framework 3 Rec. Rate
Training: Gallery; Testing: Probe			
OLS	3.69 (0.58)	4.12 (0.58)	6.45
RR	**9.56** (0.74)	9.19 (0.73)	12.10
PCR	5.00 (0.47)	6.06 (0.58)	12.90
PRO	6.25 (0.00)	6.25 (0.00)	6.45
PLS	**9.56** (0.74)	9.19 (0.73)	12.10
REM	5.62 (0.51)	6.25 (0.60)	9.68
EDE	6.25 (0.53)	6.62 (0.58)	10.48
EDET	6.12 (0.55)	6.38 (0.62)	10.48
EDEwP	9.50 (0.70)	**9.88** (0.80)	**15.32**
Training: Probe; Testing: Gallery			
OLS	11.12 (0.61)	11.12 (0.71)	7.50
RR	31.06 (0.93)	29.31 (0.87)	**19.17**
PCR	5.00 (0.42)	4.19 (0.52)	13.33
PRO	6.25 (0.00)	6.25 (0.00)	6.67
PLS	31.06 (0.93)	29.31 (0.87)	**19.17**
REM	6.75 (0.54)	2.56 (0.38)	4.17
EDE	6.31 (0.59)	8.06 (0.62)	**19.17**
EDET	6.25 (0.49)	6.25 (0.51)	6.67
EDEwP	**33.06** (0.98)	**33.81** (0.92)	16.67

3.4 Alternative Regularization

Any point $\mathbf{V}$ in the tangent space at a point $\mathbf{U}$ of $\mathcal{G}(r,s)$ should satisfy $\mathbf{U}^T\mathbf{V} = \mathbf{0}$; this indicates an alternative regularization, namely incorporating a penalty $\bar{\mathbf{x}}_*^T\boldsymbol{\beta}^{\mathrm{M}}$ so that $\boldsymbol{\beta}^{\mathrm{M}}$ will lie closer to the tangent space. Results on hold-one-person-out age estimation for various values of the regularization parameter are shown in Fig. 3.2. We take EDEwP2 as the previously defined regularization, with EDEwP1 being the alternative penalizing $||\bar{\mathbf{x}}_*^T\boldsymbol{\beta}^{\mathrm{M}}||^2$. We see that EDEwP1 is not different – and actually performs a little worse – than an OLS fit. While both RR and EDEwP2

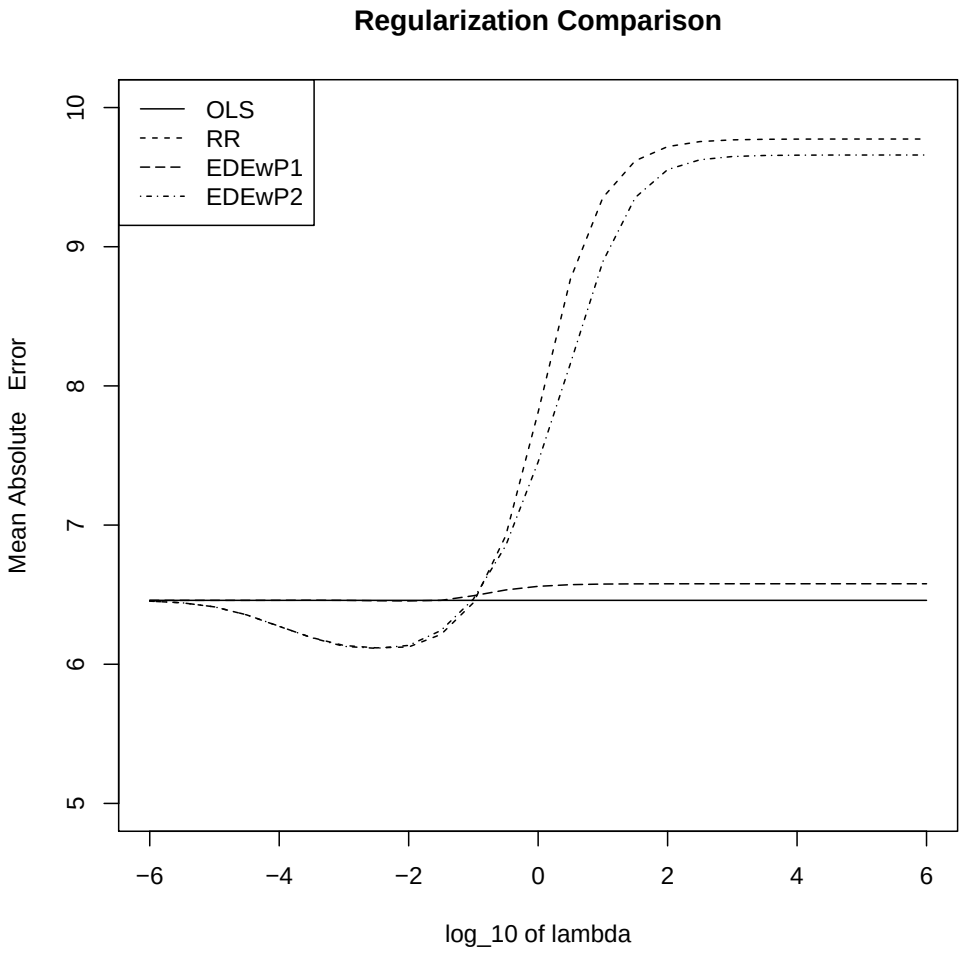


Fig. 3.2: Comparing different regularization terms.

perform well for a good choice of λ, the methods are a little more unstable. We see that ridge regression is in fact a fairly good choice for problems of this type so long as λ can be estimated well.

3.5 Discussion

By adopting an approach to incorporate structure into estimation regression parameters, both when the underlying manifold is known in advance or when the manifold is unknown, we obtain improvements in both regression and classification.

Posing the problem as an optimization and incorporating prior knowledge into the objective function results in improvements in performance and coefficient estimates that have an attractive interpretation in terms of the manifold structure of the predictors. While in some cases the data are assumed to have a globally linear structure, localization can be used to obtain better results on data that exhibit nonlinearity. We show that, although in some cases using knowledge of the prior structure does not result in large improvements, its interpretability and utility make it an indispensable tool in performing regression on manifolds.

4

Combined Direction Estimation

4.1 Introduction

As seen in the previous chapter, dimension reduction methods are a viable solution to improving predictions in the case of heterogeneous data, at least when the dimension of the predictors does not change between training and testing. A major drawback to the dimension reduction methods outlined previously is that the information about the response variable does not enter into the construction of the mapping until the end of the estimation. GFK, for example, does not use the values of the response at all except to perform nearest neighbor once we obtain the kernel. In this chapter, we propose the method of combined direction estimation (or CDE), which is closely related to the IS method in that it attempts to combine information from the source and the target in constructing a dimension-reducing transformation, though CDE may be extended to consider the conditional distribution of the response given the predictors.

4.2 Problem Setup

We assume as in Chapter 2 that we have independent predictors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ distributed as $\mathbf{X} \in \mathcal{X} \subset \mathbb{R}^p$ for training a model and $\mathbf{z}_1, \ldots, \mathbf{z}_m$ as $\mathbf{Z} \in \mathcal{Z} \subset \mathbb{R}^q$ for testing the same model. We assume we have known response variables $y_1, \ldots, y_n \in \mathbb{R}$

that are generated from some parametric process, i.e.,

$$E(Y \,|\, \mathbf{X} = \mathbf{x}_{\mathrm{i}}; \boldsymbol{\beta}) = m(\mathbf{x}_{\mathrm{i}}; \boldsymbol{\beta})$$

with $\boldsymbol{\beta} \in \mathbb{R}^p$. We use the covariate shift assumption outlined previously, namely that we have unknown response variables $\xi_1, \ldots, \xi_m$ from the the same conditional model as Y and $\mathbf{X}$, i.e., $[Y \,|\, \mathbf{X}; \boldsymbol{\beta}] \sim [\Xi \,|\, \mathbf{Z}; \boldsymbol{\beta}]$, though $[\mathbf{X}] \nsim [\mathbf{Z}]$. Our goal is to learn a model optimal under $[\Xi, \mathbf{Z}]$ while either not knowing or knowing only a small number of realizations from $[\Xi]$. Here we use $[\mathbf{X}]$ to denote the marginal distribution of random variable $\mathbf{X}$, $[Y \,|\, \mathbf{X}]$ to denote conditional distribution, and $[Y, \mathbf{X}]$ to denote joint distribution.

4.3 Methodology

We propose an approach related to the maximum likelihood approach for PCA outlined in Chapter 1 that attempts to make a connection with the intermediate subspace method described in Chapter 2. We first pose the error model for a single observation $\mathbf{X}$ as

$$\mathbf{X} = \boldsymbol{\mu}^{\mathrm{x}} + \boldsymbol{\eta}\,\boldsymbol{\nu} + \boldsymbol{\epsilon},\ \boldsymbol{\epsilon} \sim N(\mathbf{0}, \boldsymbol{\Delta}) \tag{4.1}$$

where $\boldsymbol{\epsilon}$ is a vector of random errors, $\boldsymbol{\Delta} > 0$, $\boldsymbol{\eta} \in \mathbb{R}^{p \times d}$ is a linear dimension reduction with $\boldsymbol{\eta}^T \boldsymbol{\eta} = \mathbf{I}_{\mathrm{d}}$, and $\boldsymbol{\nu}$ are corresponding unknown coefficients with $E[\boldsymbol{\nu}] = \mathbf{0}$ [64]. We pose a similar model for $\mathbf{Z}$, assuming $p = q$ and $\mathcal{X} = \mathcal{Z}$, and using the

same transformation $\boldsymbol{\eta}$, as we wish to keep the conditional distributions of the response given the predictors the same across training and testing. Our goal will be like that of sufficient dimension reduction literature [12], with the modification that instead of being interested in the conditional distributions of the response given the covariates, we focus on the marginal distributions of the features. For example, in the homogeneous case (that is, we only have data from $\mathbf{X}$), a useful dimension reduction mapping $\boldsymbol{\eta} \in \mathbb{R}^{p\times d}$ would be one that attempts to constrain $F_{\mathbf{x}}\left((\mathbf{I}_{\mathrm{p}} - \boldsymbol{\eta}\,\boldsymbol{\eta}^T)\cdot(\mathbf{x}-\boldsymbol{\mu}^{\mathrm{x}})\right)$ to be close to $F_{\mathbf{x}}(\mathbf{x}-\boldsymbol{\mu}^{\mathrm{x}})$ where $F_{\mathbf{x}}$ is a fixed cdf. If we assume

$$F_{\mathbf{x}}(\mathbf{u}) \sim N(\mathbf{0}, \sigma_x^2\,\mathbf{I}_{\mathrm{p}}),$$

we can estimate $\boldsymbol{\eta}$ through minimizing the Kullback-Leibler divergence between these distributions over $\boldsymbol{\eta}$, resulting in the well-known principal component directions. We take a similar approach, but introduce the target distribution (here called G) as well [65]. In other words, we assume $\mathcal{X} = \mathcal{Z}$ and seek an $\boldsymbol{\eta}$ constraining $G_{\mathbf{z}}(\mathbf{u}-\boldsymbol{\mu}^{\mathrm{z}})$ to be close to $F_{\boldsymbol{\eta},\mathbf{x}}(\mathbf{u}-\boldsymbol{\mu}^{\mathrm{x}})$ and simultaneously constraining $F_{\mathbf{x}}(\mathbf{u}-\boldsymbol{\mu}^{\mathrm{x}})$ to be close to $G_{\boldsymbol{\eta},\mathbf{z}}(\mathbf{u}-\boldsymbol{\mu}^{\mathrm{z}})$ where

$$F_{\boldsymbol{\eta},\mathbf{x}}(\mathbf{u}) = F_{\mathbf{x}}\left((\mathbf{I}_{\mathrm{p}} - \boldsymbol{\eta}\,\boldsymbol{\eta}^T)\cdot\mathbf{u}\right),\ G_{\boldsymbol{\eta},\mathbf{z}}(\mathbf{u}) = G_{\mathbf{z}}\left((\mathbf{I}_{\mathrm{p}} - \boldsymbol{\eta}\,\boldsymbol{\eta}^T)\cdot\mathbf{u}\right),\ \mathbf{u}\in\mathcal{X}.$$

Intuitively, we attempt to pose a dimension reduction $\boldsymbol{\eta}$ so that points from $\mathbf{x}$ get

mapped into points that "look like" points from $\mathbf{z}$ for the first constraint, with a similar intuition for the second constraint. We will rarely require one distribution to be mapped completely to the other, which is an implicit goal in the sampling of the geodesic in IS. Instead, we will seek to minimize the average between both KL-divergences, that is, to find

$$\arg\min_{\boldsymbol{\eta}} \; J(\boldsymbol{\eta}) \text{ subject to } \boldsymbol{\eta}^T \boldsymbol{\eta} = \mathbf{I}_{\mathrm{d}}$$

where

$$J(\boldsymbol{\eta}) = \frac{1}{2}[D_{KL}(F_{\mathbf{x}}||G_{\boldsymbol{\eta},\mathbf{z}}) + D_{KL}(G_{\mathbf{z}}||F_{\boldsymbol{\eta},\mathbf{x}})],$$

with the Kullback-Leibler divergence

$$D_{KL}(F_{\mathbf{x}}||G_{\mathbf{z}}) = \int_{\mathcal{X}} f(\mathbf{u}) \log \frac{f(\mathbf{u})}{g(\mathbf{u})} d\,\mathbf{u}$$

provided F and G admit density functions f and g, respectively.

4.3.1 Error Structure

Our first assumption will be that of normal, isotropic errors. We set

$$\mathbf{X} = \boldsymbol{\mu}^{\mathrm{x}} + \boldsymbol{\eta}\,\boldsymbol{\nu}_{\mathrm{x}} + \boldsymbol{\epsilon},\ \boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma_x^2\,\mathbf{I}_{\mathrm{p}}),$$

$$\mathbf{Z} = \boldsymbol{\mu}^{\mathrm{z}} + \boldsymbol{\eta}\,\boldsymbol{\nu}_{\mathrm{z}} + \boldsymbol{\epsilon},\ \boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma_z^2\,\mathbf{I}_{\mathrm{p}})$$

where σ_x^2 and σ_z^2 are chosen parameters. In this case we note

$$\begin{aligned} J(\boldsymbol{\eta}) &= \frac{1}{2}[D_{KL}(F_{\mathbf{x}}||G_{\boldsymbol{\eta},\mathbf{z}}) + D_{KL}(G_{\mathbf{z}}||F_{\boldsymbol{\eta},\mathbf{x}})] \\ &\propto -\frac{1}{2}\left[\int_{\mathcal{X}} \frac{1}{\sigma_z^2}(\mathbf{u}-\boldsymbol{\mu}^{\mathrm{z}})^T \boldsymbol{\eta}\,\boldsymbol{\eta}^T(\mathbf{u}-\boldsymbol{\mu}^{\mathrm{z}})d\,\mathbf{u}\right. \\ &\qquad \left. + \int_{\mathcal{Z}} \frac{1}{\sigma_x^2}(\mathbf{u}-\boldsymbol{\mu}^{\mathrm{x}})^T \boldsymbol{\eta}\,\boldsymbol{\eta}^T(\mathbf{u}-\boldsymbol{\mu}^{\mathrm{x}})d\,\mathbf{u}\right] \\ &\approx -\frac{1}{2}\mathrm{tr}\left\{\frac{1}{\sigma_z^2}\boldsymbol{\Sigma}^{\mathrm{x}}\boldsymbol{\eta}\,\boldsymbol{\eta}^T + \frac{1}{\sigma_x^2}\boldsymbol{\Sigma}^{\mathrm{z}}\boldsymbol{\eta}\,\boldsymbol{\eta}^T\right. \\ &\qquad \left. + \left(\frac{1}{\sigma_x^2}+\frac{1}{\sigma_z^2}\right)(\boldsymbol{\mu}^{\mathrm{x}}-\boldsymbol{\mu}^{\mathrm{z}})(\boldsymbol{\mu}^{\mathrm{x}}-\boldsymbol{\mu}^{\mathrm{z}})^T\boldsymbol{\eta}\,\boldsymbol{\eta}^T\right\} \end{aligned}$$

where $(\boldsymbol{\mu}^{\mathrm{x}}, \boldsymbol{\Sigma}^{\mathrm{x}})$ and $(\boldsymbol{\mu}^{\mathrm{z}}, \boldsymbol{\Sigma}^{\mathrm{z}})$ are the mean vectors and covariance matrices corresponding to data $\mathbb{X}$ and $\mathbb{Z}$, respectively. As our goal is to estimate

$$\arg\min_{\boldsymbol{\eta}} \; J(\boldsymbol{\eta}) \text{ subject to } \boldsymbol{\eta}^T\boldsymbol{\eta} = \mathbf{I}_{\mathrm{d}},$$

we see that the solution in this case becomes the eigenvectors corresponding to the largest d eigenvalues of the matrix

$$\mathbf{A} = \frac{1}{\sigma_z^2}\boldsymbol{\Sigma}^{\mathrm{x}} + \frac{1}{\sigma_x^2}\boldsymbol{\Sigma}^{\mathrm{z}} + \left(\frac{1}{\sigma_x^2}+\frac{1}{\sigma_z^2}\right)(\boldsymbol{\mu}^{\mathrm{x}}-\boldsymbol{\mu}^{\mathrm{z}})(\boldsymbol{\mu}^{\mathrm{x}}-\boldsymbol{\mu}^{\mathrm{z}})^T. \tag{4.2}$$

The third term is a rank one matrix and will only affect the first eigenvector of $\mathbf{A}$; for ease of exposition we will ignore it for now, as it can be dropped by mean-centering both $\mathbb{X}$ and $\mathbb{Z}$.

To make a connection with incremental subspace learning, we choose to rescale our objective so that the solution for $\boldsymbol{\eta}$ will correspond to eigenvectors of the matrix

$(1-\alpha)\,\mathbf{\Sigma}^{\mathrm{x}} + \alpha\,\mathbf{\Sigma}^{\mathrm{z}}$ where $\alpha = \sigma_z^2/(\sigma_x^2+\sigma_z^2)$, or the proportion of variance from the "target" predictors. Our estimate for σ_x^2 and σ_z^2 will come from the average of the standard deviations of each column of $\mathbb{X}$ and $\mathbb{Z}$, respectively.

We propose a distributional relaxation in the form of a nonparametric estimator for the distribution of the errors from source and target. For ease of computation, we will mainly focus on the nonparametric estimate

$$\hat{f}_n(\mathbf{x}_0) = \frac{1}{n}\sum_{i=1}^{n} K_{\mathbf{H}\mathbf{x}}[(\mathbf{I}_{\mathrm{p}} - \boldsymbol{\eta}\,\boldsymbol{\eta}^T)(\mathbf{x}_{\mathrm{i}} - \mathbf{x}_0)]$$

where $K_{\mathbf{H}\mathbf{x}}$ is the radial Gaussian kernel described previously. We consider a simplified scenario in which $\mathbf{H}_{\mathrm{x}}$ is a multiple of the identity, namely $\mathbf{H}_{\mathrm{x}} = \sigma_x\,\mathbf{I}_{\mathrm{p}}$ (similar for $\mathbf{H}_z$). In this case we have

$$\begin{aligned} J(\boldsymbol{\eta}) &= \frac{1}{2}[D_{KL}(F_{\mathbf{x}}||G_{\boldsymbol{\eta},\mathbf{z}}) + D_{KL}(G_{\mathbf{z}}||F_{\boldsymbol{\eta},\mathbf{x}})] \\ &\propto -\frac{1}{2}\left[\int_{\mathcal{X}} \log\sum_{j=1}^{m}\exp\left\{-\frac{1}{\sigma_z^2}(\mathbf{u}-\mathbf{z}_{\mathrm{j}})^T(\mathbf{I}_{\mathrm{p}} - \boldsymbol{\eta}\,\boldsymbol{\eta}^T)(\mathbf{u}-\mathbf{z}_{\mathrm{j}})\right\} d\,\mathbf{u}\right. \\ &\qquad \left. + \int_{\mathcal{Z}} \log\sum_{i=1}^{n}\exp\left\{-\frac{1}{\sigma_x^2}(\mathbf{u}-\mathbf{x}_{\mathrm{i}})^T(\mathbf{I}_{\mathrm{p}} - \boldsymbol{\eta}\,\boldsymbol{\eta}^T)(\mathbf{u}-\mathbf{x}_{\mathrm{i}})\right\} d\,\mathbf{u}\right] \end{aligned}$$

which implies

$$J(\boldsymbol{\eta}) \approx -\frac{1}{2}\left[\sum_{i=1}^{n}\log\sum_{j=1}^{m}\exp\left\{-\frac{1}{\sigma_z^2}(\mathbf{x}_\mathrm{i}-\mathbf{z}_\mathrm{j})^T(\mathbf{I}_\mathrm{p}-\boldsymbol{\eta}\,\boldsymbol{\eta}^T)(\mathbf{x}_\mathrm{i}-\mathbf{z}_\mathrm{j})\right\}\right.$$
$$\left.+\sum_{j=1}^{m}\log\sum_{i=1}^{n}\exp\left\{-\frac{1}{\sigma_x^2}(\mathbf{z}_\mathrm{j}-\mathbf{x}_\mathrm{i})^T(\mathbf{I}_\mathrm{p}-\boldsymbol{\eta}\,\boldsymbol{\eta}^T)(\mathbf{z}_\mathrm{j}-\mathbf{x}_\mathrm{i})\right\}\right]. \quad (4.3)$$

The gradient for the above objective turns out to be

$$J_{\boldsymbol{\eta}} = -\left[\frac{1}{\sigma_z^2}\sum_{i=1}^{n}\sum_{j=1}^{m}w_{ij}^z(\boldsymbol{\eta})\cdot(\mathbf{x}_\mathrm{i}-\mathbf{z}_\mathrm{j})(\mathbf{x}_\mathrm{i}-\mathbf{z}_\mathrm{j})^T\boldsymbol{\eta}\right.$$
$$\left.\frac{1}{\sigma_x^2}\sum_{j=1}^{m}\sum_{i=1}^{n}w_{ij}^x(\boldsymbol{\eta})\cdot(\mathbf{z}_\mathrm{j}-\mathbf{x}_\mathrm{i})(\mathbf{z}_\mathrm{j}-\mathbf{x}_\mathrm{i})^T\boldsymbol{\eta}\right]$$

where

$$w_{ij}^z(\boldsymbol{\eta}) = \frac{\exp\{-(\mathbf{x}_\mathrm{i}-\mathbf{z}_\mathrm{j})^T(\mathbf{I}_\mathrm{p}-\boldsymbol{\eta}\,\boldsymbol{\eta}^T)(\mathbf{x}_\mathrm{i}-\mathbf{z}_\mathrm{j})/\sigma_z^2\}}{\sum_{k=1}^{n}\exp\{-(\mathbf{x}_\mathrm{i}-\mathbf{z}_\mathrm{k})^T(\mathbf{I}_\mathrm{p}-\boldsymbol{\eta}\,\boldsymbol{\eta}^T)(\mathbf{x}_\mathrm{i}-\mathbf{z}_\mathrm{k})/\sigma_z^2\}}.$$

We take two approaches to speed up this method. First, in the kernel density estimation, we cluster the data into representative subsets so that we are not working with double sums over all data from source and target. For instance, in the first term of $J(\boldsymbol{\eta})$ in (4.3), instead of summing over all j we only sum over centroids from $\mathbf{z}$, with a similar approach being done for data from $\mathbf{x}$ in the second term. For the source data we may take within-class means, but since we do not wish to constrain ourselves to the setting of assuming response variables from the target, we forgo this for now and instead use standard clustering techniques (cf. [49]). Our second

simplification is in the estimation of $\boldsymbol{\eta}$. We use a previous (or initial) estimate of $\boldsymbol{\eta}$ in calculating weights $w_{ij}(\boldsymbol{\eta})$ so the unknown $\boldsymbol{\eta}$ does not appear in these weights. In this case we have a similar solution to the case in which we assumed the errors were normal, except now taking $\boldsymbol{\eta}$ as the largest d eigenvectors of

$$\mathbf{A} = \frac{1}{\sigma_z^2} \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij}^z \cdot (\mathbf{x}_\mathrm{i} - \mathbf{z}_\mathrm{j})(\mathbf{x}_\mathrm{i} - \mathbf{z}_\mathrm{j})^T + \frac{1}{\sigma_x^2} \sum_{j=1}^{m} \sum_{i=1}^{n} w_{ij}^x \cdot (\mathbf{z}_\mathrm{j} - \mathbf{x}_\mathrm{i})(\mathbf{z}_\mathrm{j} - \mathbf{x}_\mathrm{i})^T,$$

an analogue to the covariance matrices in the isotropic error case. If we were to use one cluster our solution would be identical to this isotropic normal error case.

4.3.2 Incorporating Conditional Model

One issue with a number of dimension reduction approaches is that $\boldsymbol{\eta}$ has no information about the source response in its construction. The incremental subspace method attempts to mitigate this problem by concatenating all intermediate representations and performing PLS to "average out" the effects of the changes in the conditional distributions that the application of this transformation has caused. We approach this problem directly through maximum likelihood by using $J(\boldsymbol{\eta})$ as a regularization term so that we seek

$$\underset{(\boldsymbol{\beta},\boldsymbol{\eta})}{\arg\min} \; E_{Y,\mathbf{X}} \mathcal{L}(Y, \boldsymbol{\eta}^T \mathbf{X}; \boldsymbol{\beta}) + \mu_0 \cdot J(\boldsymbol{\eta})$$

for $\boldsymbol{\eta}^T \boldsymbol{\eta} = \mathbf{I}_\mathrm{d}$ where $\mathcal{L}(\cdot, \cdot; \boldsymbol{\beta})$ is an appropriate loss function with parameter $\boldsymbol{\beta}$ to

be estimated and $\mu_0 > 0$ a chosen regularization parameter. This regularization is similar in spirit to the elastic net [27] in its regularization of a convex combination of objectives, though in this case the regularization seeks a useful transformation as opposed to useful properties for the model parameter $\boldsymbol{\beta}$.

For regression problems we assume

$$\mathbf{y} = \mathbb{X}\,\boldsymbol{\eta}\,\boldsymbol{\beta} + \mathbf{e}_{\mathrm{d}},\ \mathbf{e}_{\mathrm{d}} \sim N(\mathbf{0}, \sigma_e^2\,\mathbf{I}_{\mathrm{n}})$$

with the squared error loss, that is, for $m(\mathbf{X}; \boldsymbol{\beta}) = \beta_0 + \mathbf{X}^T\,\boldsymbol{\beta}_1$ and

$$\mathcal{L}(Y, \mathbf{X}; \boldsymbol{\beta}) = \frac{1}{2\sigma_e^2}(Y - \beta_0 - \mathbf{X}^T\,\boldsymbol{\beta}_1)^2.$$

To simplify analysis, we assume both $\mathbf{X}$ and $\mathbf{Z}$ have zero mean so that β_0 will not need to be estimated. Our analogue to incremental subspace learning will require the choice of tuning parameters μ_0 and α, and then will estimate $\boldsymbol{\eta}$ by minimizing the sample objective with respect to $\boldsymbol{\eta}$ while taking $\boldsymbol{\beta}$ to be the least squares solution with $\mathbf{X}\,\boldsymbol{\eta}$ and $\mathbf{Y}$. In this case, we note the gradient with respect to $\boldsymbol{\eta}$ becomes

$$L_{\boldsymbol{\eta}} = \frac{1}{\sigma_e^2}\left(\mathbb{X}^T\,\mathbb{X}\,\boldsymbol{\eta}\,\boldsymbol{\beta}\,\boldsymbol{\beta}^T - \mathbb{X}^T\,\mathbf{y}\,\boldsymbol{\beta}^T\right) + \mu_0 \cdot J_{\boldsymbol{\eta}} \tag{4.4}$$

where $J_{\boldsymbol{\eta}}$ is the gradient of J with respect to $\boldsymbol{\eta}$ and

$$\boldsymbol{\beta} = (\boldsymbol{\eta}^T\,\mathbb{X}^T\,\mathbb{X}\,\boldsymbol{\eta})^{-1}(\boldsymbol{\eta}^T\,\mathbb{X}^T\,\mathbf{y}).$$

The estimator $\hat{\boldsymbol{\eta}}$ is obtained by setting (4.4) above to zero and solving for $\boldsymbol{\eta}$. This

method of incorporating information from the conditional model is similar to an extension of the reduced rank regression problem described in [66]. The following theorem pertains to the asymptotics of the estimator $\widehat{\boldsymbol{\eta}}$.

Theorem. *Under some modeling and regularity assumptions given in Section 4.8, for $\mu_0 = o(n)$, if $\hat{\boldsymbol{\eta}}$ is the solution to setting* (4.4) *equal to zero, then*

$$\sqrt{n}[\text{vec}(\hat{\boldsymbol{\eta}}^T) - \text{vec}(\boldsymbol{\eta}^T)] \xrightarrow{D} N_{pd}(\mathbf{0}, \mathbf{V}[\boldsymbol{\eta}])$$

*where "*vec*" denotes vectorization and $\mathbf{V}(\boldsymbol{\eta})$ is a covariance matrix depending on the unknown parameter $\boldsymbol{\eta}$.*

See Section 4.8 for details.

For classification problems, we can instead use the negative log-likelihood as our loss function so that, for example in a C-class classification problem (i.e., $Y \in \{1, \ldots, C\}$), we minimize over $(\boldsymbol{\beta}, \boldsymbol{\eta})$ the function $L(\boldsymbol{\beta}, \boldsymbol{\eta})$ where

$$L(\boldsymbol{\beta}, \boldsymbol{\eta}) = -\left(\sum_y \mathbf{y}_{\text{y}}^T \, \mathbb{X} \, \boldsymbol{\eta} \, \boldsymbol{\beta}_{\text{y}}\right) + \left(\sum_i \log[1 + \sum_y \exp\{\mathbf{x}_{\text{i}}^T \, \boldsymbol{\eta} \, \boldsymbol{\beta}_{\text{y}}\}]\right) + \mu_0 \cdot J(\boldsymbol{\eta})$$

with element i of $\mathbf{y}_{\text{y}}$ as $\mathbf{1}\{y_i = y\}$. The gradient of L with respect to $\boldsymbol{\eta}$ becomes

$$L_{\boldsymbol{\eta}} = -\left(\sum_y \mathbb{X}^T \, \mathbf{y}_{\text{y}} \, \boldsymbol{\beta}_{\text{y}}^T\right) + \left(\sum_i \sum_y \frac{\exp\{\mathbf{x}_{\text{i}}^T \, \boldsymbol{\eta} \, \boldsymbol{\beta}_{\text{y}}\}}{1 + \sum_y \exp\{\mathbf{x}_{\text{i}}^T \, \boldsymbol{\eta} \, \boldsymbol{\beta}_{\text{y}}\}} \cdot \mathbf{x}_{\text{i}} \, \boldsymbol{\beta}_{\text{y}}^T\right) + \mu_0 \cdot J_{\boldsymbol{\eta}}.$$

which can similarly be solved through alternating minimization.

Unfortunately this technique is computationally expensive for high-dimensional problems. We instead apply the techniques for regression to a least squares classification model. In other words, for a C-class problem, we fit C linear models to the response $Y_k = \mathbf{1}\{Y = k\}$ for $Y \in \{1, \ldots, C\}$ and $k = 1, \ldots, C$. Refer to Table 4.3 in Section 4.5 for justification.

4.4 Prior Structure

In the case of predictors with prior structure our normal error model no longer holds on the data as given. We will now propose two approaches, similar to those outlined, but for structured data.

Our structure will come from observations $\mathbf{X} \in \mathcal{G}(r, s)$ identified with points in $\mathbb{R}^{r \times s}$ satisfying $\mathbf{X}^T \mathbf{X} = \mathbf{I}_s$. For the first method, we use a directional model, assuming data coming from a matrix Langevin distribution [2]

$$f(\mathbf{X}; \boldsymbol{\mu}, k_x) = a(k_x) \cdot \text{etr}\{k_x \cdot \boldsymbol{\mu}^{\text{x}T} \mathbf{X}\}$$

where k_x is a fixed parameter, $\text{etr}\{\cdot\}$ denotes $\exp[\text{tr}\{\cdot\}]$, and the location parameter $\boldsymbol{\mu}^{\text{x}} \in \mathbb{R}^{r \times s}$ is a basis for a point in $\mathcal{G}(r, s)$.

In this case we take $\boldsymbol{\eta} \in \mathbb{R}^{r \times d}$ with $\boldsymbol{\eta}^T \boldsymbol{\eta} = \mathbf{I}_{\text{d}}$. In other words, we are interested in predictors still spanning s-dimensional subspaces, but this time subspaces of $\mathbb{R}^d$ as opposed to $\mathbb{R}^r$. The parameter $\boldsymbol{\mu}^{\text{x}}$ can be estimated for large n and $r >> s$ as $r \cdot \bar{\mathbf{x}}$. We now see that J in this case becomes

$$J(\boldsymbol{\eta}) \approx -\left[\operatorname{tr}(k_z\,\boldsymbol{\mu}^{\mathrm{z}T}\,\boldsymbol{\eta}\,\boldsymbol{\eta}^T\,\bar{\mathbf{x}}) + \operatorname{tr}(k_x\,\boldsymbol{\mu}^{\mathrm{x}T}\,\boldsymbol{\eta}\,\boldsymbol{\eta}^T\,\bar{\mathbf{z}})\right],$$

but unfortunately we no longer have the guarantee that $\mathbf{x}_{\mathrm{i}}{}^T\,\boldsymbol{\eta}\,\boldsymbol{\eta}^T\,\mathbf{x}_{\mathrm{i}} = \mathbf{I}_{\mathrm{s}}$. That is, our reduced predictors do not necessarily span $\mathbb{R}^s$. We add a constraint for each observation so that this will hold. In this case, the optimal solution for $\boldsymbol{\eta}$ becomes the eigenvectors corresponding to the largest d eigenvalues of

$$\mathbf{A} = \left(k_z\bar{\mathbf{x}}\,\boldsymbol{\mu}^{\mathrm{z}T} + k_x\bar{\mathbf{z}}\,\boldsymbol{\mu}^{\mathrm{x}T}\right) - \lambda\left(\sum_{i=1}^{n}\mathbf{x}_{\mathrm{i}}\,\mathbf{x}_{\mathrm{i}}^T + \sum_{j=1}^{m}\mathbf{z}_{\mathrm{j}}\,\mathbf{z}_{\mathrm{j}}^T\right)$$

where $\lambda > 0$ is a regularization parameter. We will fix $k_x = k_z = 1$.

We also consider the case in which data is modeled with an appropriate kernel. The density estimate for $\mathbb{X}$ will be

$$\tilde{f}_n(\mathbf{x}_0) = \frac{1}{n}\sum_{i=1}^{n}\widetilde{K}_{\mathbf{H}\,\mathbf{x}}(\boldsymbol{\eta}^T\,\mathbf{x}_{\mathrm{i}}, \boldsymbol{\eta}^T\,\mathbf{x}_0)$$

where

$$\widetilde{K}_{\mathbf{H}\,\mathbf{x}}(\mathbf{u},\mathbf{v}) = \omega(\mathbf{H}_{\mathrm{x}})\cdot\exp\{-\operatorname{tr}[\mathbf{H}_{\mathrm{x}}^{-1}(\mathbf{I}_{\mathrm{s}} - g_P(\mathbf{u},\mathbf{v}))]\}$$

with $\mathbf{H}_{\mathrm{x}}$ an $s\times s$ bandwidth matrix and $\omega(\mathbf{H}_{\mathrm{x}})$ a normalizing factor [2]. We let

$$g_P(\mathbf{u},\mathbf{v}) = \frac{1}{2}(\mathbf{u}^T\,\mathbf{v}\,\mathbf{v}^T\,\mathbf{u} + \mathbf{v}^T\,\mathbf{u}\,\mathbf{u}^T\,\mathbf{v})$$

as in Chapter 3 because it yields a symmetric kernel. In this case we have

$$J(\boldsymbol{\eta}) = \sum_{i=1}^{n} \log \sum_{j=1}^{m} \exp\{-\frac{1}{\sigma_z} \mathrm{tr}[\mathbf{I}_\mathrm{s} - g(\boldsymbol{\eta}^T \mathbf{x}_\mathrm{i}, \boldsymbol{\eta}^T \mathbf{z}_\mathrm{j})]\} + \sum_{j=1}^{m} \log \sum_{i=1}^{n} \exp\{-\frac{1}{\sigma_x} \mathrm{tr}[\mathbf{I}_\mathrm{s} - g(\boldsymbol{\eta}^T \mathbf{z}_\mathrm{j}, \boldsymbol{\eta}^T \mathbf{x}_\mathrm{i})]\}$$

Using a similar simplification as in the case of the nonparametric model for the normal random errors, we estimate the solution for $\boldsymbol{\eta}$ as the eigenvectors corresponding to the largest d eigenvalues of the matrix

$$\mathbf{A} = \frac{1}{\sigma_z} \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij}^{x} (\mathbf{x}_\mathrm{i} \mathbf{x}_\mathrm{i}^T \boldsymbol{\eta} \boldsymbol{\eta}^T \mathbf{z}_\mathrm{j} \mathbf{z}_\mathrm{j}^T + \mathbf{z}_\mathrm{j} \mathbf{z}_\mathrm{j}^T \boldsymbol{\eta} \boldsymbol{\eta}^T \mathbf{x}_\mathrm{i} \mathbf{x}_\mathrm{i}^T) + \frac{1}{\sigma_x} \sum_{j=1}^{m} \sum_{i=1}^{n} w_{ij}^{z} (\mathbf{z}_\mathrm{j} \mathbf{z}_\mathrm{j}^T \boldsymbol{\eta} \boldsymbol{\eta}^T \mathbf{x}_\mathrm{i} \mathbf{x}_\mathrm{i}^T + \mathbf{x}_\mathrm{i} \mathbf{x}_\mathrm{i}^T \boldsymbol{\eta} \boldsymbol{\eta}^T \mathbf{z}_\mathrm{j} \mathbf{z}_\mathrm{j}^T).$$

To estimate σ_x we recall from Chapter 3 that the bias and the variance satisfy

$$b[\tilde{f}_n(\mathbf{x}_0)] = O[\mathrm{tr}(\mathbf{H}_\mathrm{x})], \ \mathrm{var}[\tilde{f}_n(\mathbf{x}_0)] = O([n|\,\mathbf{H}_\mathrm{x}\,|^{(r-s)/2}]^{-1}).$$

We again seek a diagonal bandwidth matrix $\mathbf{H}_\mathrm{x}^{(\mathrm{n})} = \sigma_x \mathbf{I}_\mathrm{s}$ that achieves a trade-off in bias and variance that will yield a minimum mean square error for $\tilde{f}_n(\mathbf{x}_0)$, yielding $\hat{\sigma}_x = (s^2 n)^{-2/(4+s(r-s))}$. Most examples considered will have $s = 2$, meaning $\hat{\sigma}_x = (4n)^{-1/r}$. This is similar to the rule of thumb for typical density estimation considering the dimension of $\mathcal{G}(r, s)$ is $s(r - s)$.

4.5 Simulation Studies

4.5.1 Alternative Methods

We compare various alternative methods to those proposed. A naïve approach is used by simply taking the estimate for $\boldsymbol{\eta}$ to be the d-column principal component estimate from $\mathbb{X}$ and $\mathbb{Z}$. This baseline method (based on principal components regression) will be labeled PCR. A similarly naïve covariate shift model (denoted CS) is considered using the same kernel and bandwidth estimator as is used for our nonparametric error model. Kernel mean matching (KMM) as described in Chapter 2 is also used with $\epsilon = 1 - n^{-1/2}$ and $W = 1000$ as in [50], but with $\sigma = 10^{-4}$.

As mentioned in Section 4.3.2, in [52] the incorporation of knowledge about the distribution of $Y \,|\, \mathbf{X}$ is done through obtaining a number of intermediate transformations $\boldsymbol{\eta}_k$ and concatenating the representations for $\mathbb{X}\,\boldsymbol{\eta}_k$ into a "full" representation. This full representation is used as input into a PLS with response $\mathbf{y}$ and a subspace of this full representation is estimated. In our implementation, we take the dimension of this PLS subspace to be the same as our estimate for d, and denote this method as IS for incremental subspaces.

We have four proposed approaches for combined direction estimation: normal isotropic error without and with the conditional model (CDE1 and CDE2) and nonparametric extensions of this (CDE3 and CDE4). We use the prior structure models in problems that call for it, and denote these methods as CDEP1 through CDEP4.

For all problems involving classification, either PLS or ridge regression are

used in a one-vs-all classifier [56]. For problems involving predictors lying on $\mathcal{G}(r, s)$, we use the method described in Section 4.4. Regression in this case was regularized using a ridge penalty, as it has been shown that this yields similar results to more theoretically-driven penalties [67].

Finally, to give an idea of the relative gains in predictive power each model has, we report results using a null model: mean($\mathbf{y}$) for regression and mode($\mathbf{y}$) for classification, denoted as "NULL". The means and standard errors of either the mean absolute errors (for regression) or the recognition rates (for classification) are reported.

4.5.2 Implementation

We conduct a simulation similar to those studies used in [68]. We generate $\mathbf{X}$ and $\mathbf{Z}$ the same as in Chapter 2 with $p = q$, but take

$$\boldsymbol{\eta}_1 = \mathbf{1}/\sqrt{p},\ \boldsymbol{\eta}_2 = (-\mathbf{1})^j/\sqrt{p},\ j = 1, \ldots, p$$

where $\mathbf{1}$ is the vector with all components equal to one and the exponent in $\boldsymbol{\eta}_2$ is taken elementwise. To incorporate a simulation study with prior structure, we also perform a similar simulation to those above for 20-dimensional variates, reshaping and orthogonalizing them so that they lie in $\mathcal{G}(10, 2)$. For these studies with prior structure, we take

$$Y = \mathbf{1}^T \boldsymbol{\eta}_1^T \mathbf{X} - \mathbf{1}^T \boldsymbol{\eta}_2^T \mathbf{X} + \epsilon,$$

Table 4.1: Regression simulation. The averages and standard errors of mean absolute errors (MAE) are calculated after 100 replications, with minima given in bold.

Method	Unstructured Data		Structured Data	
	Source MAE (SE)	Target MAE (SE)	Source MAE (SE)	Target MAE (SE)
NULL	2.928 (0.016)	6.122 (0.007)	1.741 (0.009)	8.451 (0.054)
PCR	1.211 (0.008)	1.559 (0.007)	0.914 (0.005)	1.504 (0.011)
CS	1.214 (0.008)	1.572 (0.007)	0.918 (0.005)	1.512 (0.011)
KMM	1.211 (0.008)	1.559 (0.007)	0.921 (0.005)	1.551 (0.014)
IS	1.265 (0.010)	1.641 (0.012)	1.327 (0.016)	4.380 (0.099)
CDE1	1.251 (0.010)	1.563 (0.008)	0.915 (0.005)	1.505 (0.011)
CDE2	**1.043** (0.009)	**1.339** (0.008)	0.818 (0.007)	1.406 (0.013)
CDE3	1.280 (0.018)	1.586 (0.014)	0.917 (0.005)	1.478 (0.013)
CDE4	1.205 (0.012)	1.525 (0.013)	0.872 (0.005)	1.428 (0.011)
CDEP1	—	—	1.125 (0.014)	2.746 (0.083)
CDEP2	—	—	0.814 (0.007)	**1.356** (0.018)
CDEP3	—	—	0.962 (0.008)	1.623 (0.021)
CDEP4	—	—	**0.807** (0.007)	1.423 (0.022)

with similarly defined $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ for $\mathbf{X} \in \mathbb{R}^{10\times 2}$. These are used in both regression and classification settings, where for classification we discretize Y at its one-third and two-thirds quantiles.

For each study, we generate $n = 200$ observations for $\mathbb{X}$ and $m = 300$ observations for $\mathbb{Z}$ and replicate this 100 times. We train the model on half of the $\mathbb{X}$ and $\mathbb{Z}$ data and test on the remaining half. The results from each study are given in Tables 4.1 and 4.2. Looking at the results on the target data, in the case of regression, CDE2 and CDEP2 perform best given unstructured and structured data, respectively. In the case of classification, CDE2 and CDE4 seem to perform best on the target data, indicating that perhaps incorporating prior structure into the model may not have much benefit.

Multinomial logistic models in the case of classification are especially com-

Table 4.2: Classification simulation. The averages and standard errors of the misclassification rates (MR) in percentage points are calculated after 100 replications, with minima given in bold.

Method	Unstructured Data		Structured Data	
	Source MR (SE)	Target MR (SE)	Source MR (SE)	Target MR (SE)
NULL	69.98 (0.17)	63.08 (1.37)	70.26 (0.19)	67.51 (1.77)
PCR	31.72 (0.34)	12.03 (0.10)	39.58 (0.40)	17.85 (0.26)
CS	31.74 (0.36)	12.17 (0.09)	39.87 (0.36)	18.55 (0.28)
KMM	31.71 (0.34)	12.03 (0.10)	43.23 (0.46)	18.48 (0.33)
IS	32.27 (0.50)	11.68 (0.10)	39.23 (0.41)	19.26 (0.27)
CDE1	32.19 (0.34)	11.96 (0.10)	39.25 (0.40)	17.83 (0.26)
CDE2	**30.14** (0.36)	**11.17** (0.10)	36.81 (0.50)	**17.07** (0.27)
CDE3	32.61 (0.37)	11.61 (0.10)	39.41 (0.44)	18.03 (0.27)
CDE4	**30.14** (0.36)	**11.17** (0.10)	**36.71** (0.50)	17.11 (0.27)
CDEP1	—	—	58.30 (0.55)	40.25 (1.50)
CDEP2	—	—	40.25 (0.45)	22.82 (0.99)
CDEP3	—	—	44.14 (0.49)	20.21 (0.51)
CDEP4	—	—	38.70 (0.43)	17.73 (0.40)

Table 4.3: Means of recognition rates and computation times from multinomial logit and least squares classifier models.

	Logit		Least Squares	
	Rec. Rate	Runtime (seconds)	Rec. Rate	Runtime (seconds)
$d = 3$	85.57	12.13	78.40	0.05
$d = 6$	73.18	22.29	65.52	0.11
$d = 9$	62.87	32.66	59.28	0.16
$d = 12$	53.39	41.91	53.99	0.23
$d = 15$	47.20	51.15	52.19	0.29
$d = 18$	41.81	60.28	50.05	0.35
$d = 21$	37.25	72.66	48.66	0.42
$d = 24$	35.99	82.32	48.61	0.50
$d = 27$	34.14	94.60	48.18	0.59

putationally intensive; see Table 4.3. The above simulation study for predictors without structure was run for various values of the ambient dimension, with the average of the recognition rates and computation times reported. As seen in the table, while multinomial logit models perform well for lower-dimensional predictors,

in most cases the computational complexity outweighs the added benefit of better predictions, with least squares performing better for higher dimensions regardless. This will be used as justification for the least squares classifier in Section 4.6.

4.6 Case Studies

In addition to the previous simulation studies, we consider various case studies. The alternative methods used will be the same as those from the simulation studies. In all tests, we perform ten replications in which half of the observations from the source (as well as target) are chosen randomly without replacement as training data and test on the remaining data.

4.6.1 Diabetes Data

We consider the diabetes dataset from [69]. The data consists of 442 records of six serum measurements along with the attributes of age, sex, blood pressure, and BMI, with interest being in predicting the response, a quantitative measure of an individual's disease progression one year after baseline. Dealing with such data, interest often lies in finding a useful explanatory model given a large number of features to consider. We expand the set of predictors by considering all second-order terms of the continuous predictors (i.e., we remove sex from consideration in all models).

It may be the case that we only have access to data from a certain demographic on which to train a model. We consider two examples of source data: individuals less

Table 4.4: Means and standard errors of mean absolute errors on the diabetes data. The estimated dimension was taken to be 5. Minimum mean absolute errors are in bold.

Method	**Source: Age Less Than 50**		**Source: Males**	
	Source MAE (SE)	**Target MAE (SE)**	**Source MAE (SE)**	**Target MAE (SE)**
NULL	135.04 (0.97)	167.96 (0.78)	138.31 (1.63)	166.82 (0.60)
PCR	63.28 (5.76)	72.31 (7.74)	62.12 (5.03)	70.46 (6.50)
CS	58.26 (2.46)	64.24 (3.76)	59.45 (3.39)	67.06 (4.83)
KMM	65.26 (5.70)	72.71 (7.73)	62.44 (4.96)	72.65 (6.47)
IS	60.62 (5.82)	69.50 (8.01)	69.53 (8.22)	79.35 (9.32)
CDE1	65.70 (6.45)	76.41 (7.88)	74.25 (6.59)	87.36 (9.22)
CDE2	**53.26** (1.54)	**61.57** (2.48)	**52.33** (2.34)	**59.91** (2.38)
CDE3	71.47 (5.77)	85.80 (8.47)	71.21 (7.95)	84.02 (10.55)
CDE4	69.23 (6.51)	82.32 (9.39)	79.55 (9.12)	91.88 (12.15)

than 50 years of age; and individuals who are male. In the first case, we note that the source data of individuals under the age of 50 has 227 observations, while the source data in the second case has 235 observations. In the second case, we include age as a predictor, resulting in 54 variables (as opposed to 44 for the first case). As it may be beneficial for interpretation to have a lower-dimensional explanatory model, we fix the estimated data dimension at 5. For both tests we randomly split the data in half and run the analysis ten times.

Results are given in Table 4.4. We see that the CS and CDE2 methods perform best when the source data is individuals less than 50 years of age, while CDE2 performs best when the source data is males. IS performs competitively in the first case, but with high variability, which may be undesirable. Often, however, practitioners will be interested in variable selection. In this case, it may be possible for the CDE method to be extended to yield sparse estimates. See Section 4.7 for details.

Table 4.5: Means and standard errors for recognition rates in object recognition. The estimated dimension was taken to be 30. Maximum recognition rates are given in bold. Source data for all experiments is taken to be webcam data.

Method	Target: Amazon		Target: DSLR		Target: Webcam	
	Source	Target	Source	Target	Source	Target
NULL	4.30 (0.22)	3.29 (0.30)	4.40 (0.24)	3.35 (0.06)	4.20 (0.28)	5.13 (0.29)
PCR	34.10 (0.46)	27.63 (0.72)	33.44 (0.56)	12.71 (0.18)	33.07 (0.94)	38.77 (0.76)
CS	19.47 (0.56)	15.50 (0.86)	20.25 (1.09)	7.18 (0.26)	31.51 (0.94)	31.96 (0.67)
KMM	30.65 (0.78)	26.47 (0.70)	27.46 (1.05)	12.00 (0.31)	33.49 (0.95)	37.04 (0.73)
IS	32.69 (0.19)	26.43 (0.50)	33.74 (0.97)	12.28 (0.18)	32.81 (0.92)	40.43 (0.83)
CDE1	33.89 (0.50)	27.87 (0.82)	32.91 (0.65)	12.59 (0.26)	33.07 (0.96)	38.62 (0.76)
CDE2	**39.92** (0.59)	**33.57** (1.03)	**42.36** (0.66)	**13.47** (0.23)	39.42 (0.90)	51.53 (0.79)
CDE3	33.32 (0.63)	26.75 (0.70)	33.17 (0.54)	12.77 (0.19)	32.79 (0.99)	38.79 (0.70)
CDE4	39.70 (0.63)	**33.57** (0.95)	42.14 (0.64)	13.43 (0.26)	**39.45** (0.91)	**51.76** (0.83)

4.6.2 Object Recognition

A typical example of a domain shift in a computer vision problem can be found in data from [4]. This dataset contains 4110 observations, with each observation an image of one of 31 different objects. Each image was taken either from the website `amazon.com`, taken with a higher-quality DSLR camera, or taken with a lower-quality webcam. We used the HOG feature extraction method described in Chapter 3 with 8 bins on 8×8 patches to extract feature vectors of length 512. We standardize the predictors so that each column has zero mean and standard deviation one.

Experiments were done by taking the webcam data as source and all remaining domains as target data. We provide the recognition rates on both source and target data in Table 4.5. We see both CDE2 and CDE4 performing best out of all methods, with CDE4 doing well when there is no difference between the distributions of the source and target data. In all cases these two methods perform much better than the alternatives, indicating that incorporating information from the conditional distribution in all stages of dimension reduction is beneficial to prediction.

Table 4.6: Means and standard errors for recognition rates on face recognition across aging with landmark points as features. The value of d was taken to be 10. Maximum recognition rates are in bold.

Method	**Source**	**Target**
	Rec. Rate (SE)	**Rec. Rate (SE)**
NULL	1.39 (0.17)	0.52 (0.11)
PCR	5.79 (0.88)	3.85 (0.65)
CS	3.52 (0.32)	3.56 (0.53)
KMM	5.79 (0.88)	3.85 (0.65)
IS	8.75 (0.45)	5.41 (0.51)
CDEP1	4.44 (0.41)	3.41 (0.40)
CDEP2	12.55 (1.79)	**9.11** (1.44)
CDEP3	3.98 (0.27)	3.26 (0.48)
CDEP4	**12.73** (1.82)	**9.11** (1.39)

4.6.3 Face Recognition Across Aging

Face recognition across aging is an example of a problem that requires adaptation to a continuous domain shift. We pose the problem on the FG-NET database described in Chapter 1. Each observation was taken to be the normalized landmark points, meaning predictors came from the Grassmannian $\mathcal{G}(68, 2)$. The source domain was taken to be those individuals who were 18 years of age and under, and the target as those individuals who were greater than 18 years of age. We report the recognition rates from replicating the studies 10 times.

We see in Table 4.6 that CDEP4 performs best while CDEP2 performs competitively. Though no method performs well on either source or target data due to the limited information given from landmark points, all alternative methods perform significantly worse than the proposed. Improvements can possibly be made by expanding the predictor vector to include more refined face information.

Table 4.7: Means and standard errors of mean absolute errors on age estimation with a geometric domain shift. The value for d was taken to be 30. Minimum mean absolute errors are in bold.

Method	Source: Centered		Source: Rotated	
	Source MAE (SE)	Target MAE (SE)	Source MAE (SE)	Target MAE (SE)
NULL	9.72 (0.09)	9.61 (0.08)	9.70 (0.06)	9.92 (0.09)
PCR	6.33 (0.09)	61.14 (3.12)	**6.41** (0.05)	6.49 (0.06)
CS	6.82 (0.11)	27.53 (2.82)	7.62 (3.89)	8.71 (0.20)
KMM	6.40 (0.09)	58.47 (3.70)	6.82 (0.08)	6.71 (0.07)
IS	**6.31** (0.09)	60.57 (3.83)	6.42 (0.05)	**6.47** (0.06)
CDEP1	8.76 (0.11)	8.43 (0.09)	8.64 (0.09)	8.67 (0.09)
CDEP2	8.41 (0.14)	**8.39** (0.13)	8.48 (0.12)	8.57 (0.09)
CDEP3	8.76 (0.11)	8.43 (0.09)	8.64 (0.09)	8.67 (0.09)
CDEP4	8.44 (0.17)	8.48 (0.14)	8.46 (0.14)	8.59 (0.15)

4.6.4 Age Estimation

Finally, to illustrate the effect a geometric change will have on each of the methods, we turn again to age estimation on the FG-NET dataset. Regression is performed on $\sqrt{y}$ where y is the individual's age as in Chapter 3 and [67]. We use the mean absolute error (MAE) for the predicted ages to measure the performance of each method.

In the tests, we randomly split the data in two, and for half of the data, for each point we rotate the landmarks by a random angle sampled uniformly between 0 and $\pi/16$ radians. Table 4.7 shows results for both using centered predictors as source data and using rotated predictors as source data. In the case in which centered predictors are used as source, most methods fail spectacularly, with the only reasonable estimates being those from the null model and the proposed methods. This illustrates an issue with many of the existing domain adaptation approaches: the methods perform quite poorly when the support of the source data does not

contain that of the target data. Here CDEP1 through CDEP4 all perform similarly, though much better than the alternatives.

When we reverse the scenario we see much better performance in the alternative methods, with the IS method performing very well on the target data. For this example, the proposed methods see similar performances in both cases, and are thus more stable with respect to the distribution of the target data points.

4.7 Extension: Sparse Estimates

A benefit to formulating our dimension reduction in terms of an objective function to be minimized is that sparsity penalties can be incorporated. For example, in cases such as the diabetes data, practitioners are often interested in both variable selection as well as dimension reduction. The current CDE method does not work well for variable selection as its output takes linear combinations of every variable under consideration. However, we can simply add an ℓ_1 penalty to the parameters to encourage many of the coefficients to be zero.

As in sparse principal component analysis [70], we note that the jth principal component of the data $\mathbb{X}$ can be obtained by solving

$$\arg\min_{\boldsymbol{\eta}_\mathrm{j}} || \mathbf{X} - \mathbf{X}\,\boldsymbol{\eta}_\mathrm{j}\,\boldsymbol{\eta}_\mathrm{j}^T ||^2 + \lambda || \boldsymbol{\eta}_\mathrm{j} ||^2$$

where the last term corresponds to the constraint $\boldsymbol{\eta}^T\,\boldsymbol{\eta} = \mathbf{I}_\mathrm{d}$. We can reformulate this as a LASSO-type problem by noting that, if $\boldsymbol{\eta}_\mathrm{j}^0$ is some initial estimate for $\boldsymbol{\eta}_\mathrm{j}$, then $\boldsymbol{\eta}_\mathrm{j}^T\,\boldsymbol{\eta}_\mathrm{j}^0 \approx 1$. In this case we can write the optimization as

Table 4.8: Elements of $\boldsymbol{\eta}$ greater than .01 in absolute value after sparse CDE.

- First Direction: $\boldsymbol{\eta}_3, \boldsymbol{\eta}_{25}$
- Second Direction: $\boldsymbol{\eta}_2, \boldsymbol{\eta}_3, \boldsymbol{\eta}_5, \boldsymbol{\eta}_{23}, \boldsymbol{\eta}_{32}, \boldsymbol{\eta}_{42}$
- Third Direction: $\boldsymbol{\eta}_{20}, \boldsymbol{\eta}_{22}$
- Fourth Direction: $\boldsymbol{\eta}_2, \boldsymbol{\eta}_8$
- Fifth Direction: $\boldsymbol{\eta}_7, \boldsymbol{\eta}_{42}$

$$\arg\min_{\boldsymbol{\eta}_\mathrm{j}} || \mathbf{X}\, \boldsymbol{\eta}_\mathrm{j}^0 - \mathbf{X}\, \boldsymbol{\eta}_\mathrm{j} ||^2 + \lambda || \boldsymbol{\eta}_\mathrm{j} ||^2 \tag{4.5}$$

which can easily have an ℓ_1 penalty incorporated directly via an elastic net [27]. The optimization in (4.5) can be solved through an iterative procedure. As a example fit, we consider the diabetes example with source data as individuals younger than 50 years of age and take $d = 5$. We set $\boldsymbol{\eta}^0$ as the top d eigenvectors of (4.2) and take a first approximation by running the elastic net above for one iteration. Table 4.8 shows the elements obtained from this approximation that are greater than .01 in absolute value. The target MAE for this sparse estimate was 50.7741 compared with 51.4834 for PCR and 51.4782 for CDE.

4.8 Choice of Regularization Parameter Rates

We assume the data $(\mathbf{y}, \mathbb{X}) \in \mathbb{R}^n \times \mathbb{R}^{n \times p}$ satisfy

$$\mathbf{y} = \mathbb{X}\, \boldsymbol{\beta}_0 + \mathbf{e}_\mathrm{p}, \;\; \mathbf{y} = \mathbb{X}\, \boldsymbol{\eta}\, \boldsymbol{\beta} + \mathbf{e}_\mathrm{d} \tag{4.6}$$

where $\mathbf{e}_{\mathrm{p}}, \mathbf{e}_{\mathrm{d}}$ are vectors of random errors with $E(\mathbf{e}_{\mathrm{p}}) = \mathbf{0}$ and $\mathrm{var}(\mathbf{e}_{\mathrm{p}}) = \sigma_p^2 \mathbf{I}_n$ with $\sigma_p^2 < \infty$ (similar for $\mathbf{e}_{\mathrm{d}}$ with σ_d^2) and $\mathrm{cov}(\mathbf{e}_{\mathrm{p}}, \mathbf{e}_{\mathrm{d}}) = \mathbf{0}$. The assumption of uncorrelated errors is made to simplify analysis and will not necessarily be true in practice. By rearranging the above two equations, we see that

$$\mathbb{N}\,\boldsymbol{\beta} - \mathbb{X}\,\boldsymbol{\beta}_0 = \mathbf{e}_{\mathrm{pd}}$$

where $\mathbb{N}$ is the matrix of coefficients $\boldsymbol{\nu}$ from the error model (4.1), $E(\mathbf{e}_{\mathrm{pd}}) = \mathbf{0}$ and $\mathrm{var}(\mathbf{e}_{\mathrm{pd}}) = (\sigma_p^2 + \sigma_d^2)\,\mathbf{I}_n$. The estimate for $\boldsymbol{\eta}^T$ that minimizes the sum of squared errors in this case for a fixed $\boldsymbol{\beta}$ and $\boldsymbol{\beta}_0$ is given by

$$\boldsymbol{\eta}^T = (\boldsymbol{\beta}\,\boldsymbol{\beta}^T)^{-1}\,\boldsymbol{\beta}\,\boldsymbol{\beta}_0^T\,.$$

Unfortunately, this solution assumes both $\mathbb{X}^T\,\mathbb{X}$ and $\boldsymbol{\beta}\,\boldsymbol{\beta}^T$ are nonsingular. This second assumption is invalid for $d > 1$ since $\boldsymbol{\beta}\,\boldsymbol{\beta}^T$ will only have rank one. Typically, due to collinearities in the high-dimensional $\mathbb{X}$, the first assumption will also not be valid as $\mathbb{X}^T\,\mathbb{X}$ will not be full rank. To overcome these issues, in estimating $\boldsymbol{\eta}^T$ we consider minimizing

$$\hat{\boldsymbol{\eta}}^T = \arg\min_{\boldsymbol{\eta}^T} ||\,\mathbf{y} - \mathbb{X}\,\boldsymbol{\eta}\,\boldsymbol{\beta}\,||^2 + \mu_1 \cdot ||\,\mathbb{X}\,\boldsymbol{\eta}\,||^2 + \mu_2 \cdot ||\,\boldsymbol{\eta}\,\boldsymbol{\beta}\,||^2 + \mu_1\mu_2 \cdot ||\,\boldsymbol{\eta}\,||^2 \qquad (4.7)$$

with $\mu_1, \mu_2 > 0$ to obtain the "least squares solution." Proceeding with straightforward calculus yields

$$\hat{\boldsymbol{\eta}}^T = (\boldsymbol{\beta}\,\boldsymbol{\beta}^T + \mu_1\,\mathbf{I}_{\mathrm{d}})^{-1}\,\boldsymbol{\beta}\,\hat{\boldsymbol{\beta}}_0^{\,T}$$

given $\boldsymbol{\beta}$ where

$$\hat{\boldsymbol{\beta}}_0 = (\mathbb{X}^T\,\mathbb{X} + \mu_2\,\mathbf{I}_{\mathrm{p}})^{-1}\,\mathbb{X}^T\,\mathbf{y}\,.$$

Solving (4.7) for $\boldsymbol{\beta}$ gives the estimate

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{\eta}^T\,\mathbb{X}^T\,\mathbb{X}\,\boldsymbol{\eta} + \mu_2\,\mathbf{I}_{\mathrm{d}})^{-1}\,\boldsymbol{\eta}^T\,\mathbb{X}^T\,\mathbf{y}$$

which we will use as a plug-in estimate for $\boldsymbol{\beta}$ to estimate $\hat{\boldsymbol{\eta}}^T$. If we let $\mu_1 = o(1)$ and fix $\mu_2 > 0$, then $\hat{\boldsymbol{\eta}}^T \to (\boldsymbol{\beta}\,\boldsymbol{\beta}^T + \mu_2\,\mathbf{I}_{\mathrm{d}})^{-1}\,\boldsymbol{\beta}\,\boldsymbol{\beta}_0^T$ as $n \to \infty$.

We now wish to optimize over the objective function in (4.7) while incorporating the penalty $\mu_0 \cdot J(\boldsymbol{\eta})$ as in (4.4). Setting the gradient equal to zero and simplifying gives

$$\hat{\boldsymbol{\eta}}_*^T + \mu_0 \cdot (\hat{\boldsymbol{\beta}}\hat{\boldsymbol{\beta}}^T + \mu_1\,\mathbf{I}_{\mathrm{d}})^{-1}\,\dot{J}^T(\boldsymbol{\eta}_*)(\mathbb{X}^T\,\mathbb{X} + \mu_2\,\mathbf{I}_{\mathrm{p}})^{-1} = \hat{\boldsymbol{\eta}}^T$$

where we seek the solution $\hat{\boldsymbol{\eta}}_*^T$. If $\mathbb{X}$ and $\mathbb{Z}$ have finite second moments, $\dot{J}$ will converge to a constant as $n, m \to \infty$. Thus, letting $\mu_0 = o(n)$ will ensure the second term goes to zero as $n \to \infty$. The following theorem holds.

Theorem. *Under the model* (4.6) *and assuming* $\mu_0, \mu_2 = o(n)$, $\mu_1 = o(1)$, *and* $\mathbf{X}, \mathbf{Z}$ *have finite second moments, the solution*

$$\begin{aligned}\hat{\boldsymbol{\eta}}^T &= \arg\min_{\boldsymbol{\eta}^T} || \mathbf{y} - \mathbb{X}\,\boldsymbol{\eta}\,\boldsymbol{\beta} ||^2 + \mu_0 \cdot J(\boldsymbol{\eta}) \\ &+ \mu_1 \cdot || \mathbb{X}\,\boldsymbol{\eta} ||^2 + \mu_2 \cdot || \boldsymbol{\eta}\,\boldsymbol{\beta} ||^2 + \mu_1\mu_2 \cdot || \boldsymbol{\eta} ||^2 \end{aligned} \tag{4.8}$$

is consistent and

$$\sqrt{n}[\mathrm{vec}(\hat{\boldsymbol{\eta}}) - \mathrm{vec}(\boldsymbol{\eta})] \xrightarrow{D} N_{pd}(\mathbf{0}, \mathbf{V}[\boldsymbol{\eta}])$$

as $n \to \infty$.

Proof. For $\mu_0 = o(n)$, as $n \to \infty$ we have

$$\hat{\boldsymbol{\eta}}^T = (\hat{\boldsymbol{\beta}}\hat{\boldsymbol{\beta}}^T + \mu_1\,\mathbf{I}_\mathrm{d})^{-1}\hat{\boldsymbol{\beta}}\hat{\boldsymbol{\beta}}_0{}^T,$$

which, if we let $\mu_1 = o(1)$ and $\mu_2 = o(n)$, converges in probability to $\boldsymbol{\eta}^T = (\boldsymbol{\beta}\,\boldsymbol{\beta}^T)^{-1}\,\boldsymbol{\beta}\,\boldsymbol{\beta}_0^T$. Let $\hat{\boldsymbol{\beta}} = (\hat{\boldsymbol{\beta}}^T, \hat{\boldsymbol{\beta}}_0{}^T)^T$ and $\boldsymbol{\beta} = (\boldsymbol{\beta}^T, \boldsymbol{\beta}_0^T)^T$ be vectors in $\mathbb{R}^{d+p}$. Then

$$\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \xrightarrow{D} N(\mathbf{0}, \mathbf{B})$$

where

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix},$$

$$\mathbf{B}_{11} = \sigma_d^2(\boldsymbol{\eta}^T\,\mathbb{X}^T\,\mathbb{X}\,\boldsymbol{\eta} + \mu_2\,\mathbf{I}_\mathrm{d})^{-1}\,\boldsymbol{\eta}^T\,\mathbb{X}^T\,\mathbb{X}\,\boldsymbol{\eta}(\boldsymbol{\eta}^T\,\mathbb{X}^T\,\mathbb{X}\,\boldsymbol{\eta} + \mu_2\,\mathbf{I}_\mathrm{d})^{-1},$$

$$\mathbf{B}_{22} = \sigma_p^2(\mathbb{X}^T\,\mathbb{X} + \mu_2\,\mathbf{I}_{\mathrm{p}})^{-1}\,\mathbb{X}^T\,\mathbb{X}(\mathbb{X}^T\,\mathbb{X} + \mu_2\,\mathbf{I}_{\mathrm{p}})^{-1},\ \mathbf{B}_{12} = \mathbf{B}_{21}^T = \mathbf{0}_{d\times p}.$$

If we let $\mathbf{G}_{\boldsymbol{\eta}}(\boldsymbol{\beta}) \in \mathbb{R}^{pd\times(d+p)}$ be the gradient of the mapping

$$\boldsymbol{\eta}^T(\boldsymbol{\beta}) = (\boldsymbol{\beta}\,\boldsymbol{\beta}^T + \mu_1\,\mathbf{I}_{\mathrm{d}})^{-1}\,\boldsymbol{\beta}\,\boldsymbol{\beta}_0^T$$

with respect to $\boldsymbol{\beta}$, then applying the delta method to $\mathrm{vec}[\boldsymbol{\eta}^T(\hat{\boldsymbol{\beta}})]$ about $\boldsymbol{\beta}$ yields

$$\sqrt{n}[\mathrm{vec}(\hat{\boldsymbol{\eta}}^T) - \mathrm{vec}(\boldsymbol{\eta}^T)] \xrightarrow{D} N(\mathbf{0}, \mathbf{V}(\boldsymbol{\eta}))$$

where $\mathbf{V}(\boldsymbol{\eta}) = \mathbf{G}_{\boldsymbol{\eta}} \cdot \mathbf{B} \cdot \mathbf{G}_{\boldsymbol{\eta}}^T$. □

4.9 Discussion

We have shown the benefits of posing the intermediate subspace approach to domain adaptation as an optimization problem. Our approach admits a general objective function to be used, and has the ability to give a more intuitive idea of what the intermediate spaces mean in terms of the data. Obtaining an intermediate space in this fashion results in improvements in both classification and regression. Furthermore, this approach is superior to alternative approaches when faced with a regression problem in which the source and target may differ by some geometric transformation. This method can be easily extended to semisupervised problems, and though it was not considered, it can handle $p \gg n$ problems as well.

We have also shown that solving a regularized optimization can yield parameter estimates with attractive statistical properties. Incremental learning to account

for shifts in domain is a useful, easily extensible practice, and attempting to solve the problem via regularization results in many improvements.

5

Regularized Likelihood Directions

5.1 Introduction

The dimension reduction methods discussed so far have suffered from a few drawbacks, chief among them the fact that they cannot naturally handle cases in which predictors arise from spaces of differing dimension, nor do they easily incorporate information pertaining to the response variable into the dimension reduction transformation. CDE attempts to overcome this second drawback by penalizing the likelihood function with a term involving a dimension reduction parameter, though this technique still essentially operates only on the conditional model $Y|\mathbf{X}$. In the following chapter we seek a method to regularize a likelihood function of the joint distribution $(Y, \mathbf{X})$ that will also be able to handle cases in which predictors differ in dimension from training to testing.

5.2 Background

5.2.1 Problem Setup

As in Chapters 2 and 4, we assume access to independent predictors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ where each $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^p$ is distributed as the random variable $\mathbf{X}$ and has a corresponding label from Y which is either continuous ($y_i \in \mathbb{R}$) or discrete ($y_i \in$

$\{1, \ldots, C\}$). The method will be described for discrete y_i but can be extended to continuous y_i via thresholding. Additionally, we have a small amount of independent predictors $\mathbf{z}_1, \ldots, \mathbf{z}_m$ with $\mathbf{z}_j \in \mathcal{Z} \subset \mathbb{R}^q$ distributed as $\mathbf{Z}$ with corresponding labels from Ξ.

We make a distributional assumption [64] on the conditional distributions $[\mathbf{X}\,|Y = y]$ and $[\mathbf{Z}\,|\Xi = y]$ in the form of

$$[\mathbf{X}\,|Y = y] \sim N(\boldsymbol{\mu}_{\mathrm{y}}^{\mathrm{x}}, \boldsymbol{\Sigma}_{\mathrm{y}}^{\mathrm{x}}),$$

$$[\mathbf{Z}\,|\Xi = y] \sim N(\boldsymbol{\mu}_{\mathrm{y}}^{\mathrm{z}}, \boldsymbol{\Sigma}_{\mathrm{y}}^{\mathrm{z}})$$

for $y = 1, \ldots, C$. In other words, we assume that within-class the features are distributed normally with differing means and covariances.[1]

Previously, as well as in typical domain adaptation literature, data from $[\mathbf{X}]$ was called "source" or "training" data, while data from $[\mathbf{Z}]$ was called "target" or "testing" data. As we assume knowledge of some data from $[\mathbf{Z}]$ at the training phase of our procedure, we use the former terminology. Our goal is again to estimate a model optimal under $[\Xi, \mathbf{Z}]$ while only knowing a small number of observations from this distribution, in this case by estimating a dimension reduction subspace by investigating the behavior of $[\mathbf{X}\,|Y]$ defined above. Specifically, we will estimate parameters $\boldsymbol{\eta} \in \mathbb{R}^{p \times d}$ and $\boldsymbol{\gamma} \in \mathbb{R}^{q \times d}$ where we assume d is fixed in advance and that $p \neq q$ in general. Typically we assume $p \gg n$; even for $p < n$, data are often assumed to be manifold-valued, so it is usually the case that $\mathrm{cov}(\mathbf{X})$ (here called

[1] N.B. $[\mathbf{X}\,|Y]$ and $[\mathbf{Z}\,|\Xi]$ will not be similarly distributed under the covariate shift assumptions.

Original images Reduced images

Fig. 5.1: Sample images from `amazon.com` (top), webcam (middle), and a DSLR camera (bottom). Reduced images are scaled to 100×100 from 10×10 for visualization. Images taken from [4].

$\boldsymbol{\Sigma}^{\mathrm{x}}$) is singular.

5.2.2 Sufficient Dimension Reduction

A number of domain adaptation methods seek optimal linear transformations to reduce the dimension of the data, though "optimality" is often largely problem-dependent. Methods minimizing reconstruction error objectives [e.g., principal component analysis (PCA)] are popular, though often we desire a transformation that takes into account additional information, such as labels of the response in a classification problem. In the statistics literature, sliced inverse regression (SIR) [71] attempts to incorporate this information through the within-class first moments. Estimates for $\boldsymbol{\eta}$ are obtained as the top d eigenvectors of $(\boldsymbol{\Sigma}^{\mathrm{x}})^{-1}\,\mathbf{M}\,\mathbf{M}^T$ where $\mathbf{M}$ is the $\mathbb{R}^{p\times C}$ matrix of within-class means.

Sufficient dimension reduction [12] – in which a transformation $\boldsymbol{\eta}$ is esti-

mated so that $[Y \,|\, \boldsymbol{\eta}^T \mathbf{X}] \sim [Y \,|\, \mathbf{X}]$ – is closely related to the method of SIR above. Likelihood-acquired directions (LAD, [64]), a type of sufficient dimension reduction method, are estimated by maximizing

$$L(\boldsymbol{\eta}; \mathbb{X}, \mathbf{y}) = \frac{1}{2} \log |\, \boldsymbol{\eta}^T \boldsymbol{\Sigma}^{\mathrm{x}} \boldsymbol{\eta} \,| - \frac{1}{2} \sum_{y=1}^{C} \frac{n_y}{n} \log |\, \boldsymbol{\eta}^T \boldsymbol{\Sigma}^{\mathrm{x}}_{\mathrm{y}} \boldsymbol{\eta} \,| \tag{5.1}$$

over all $\boldsymbol{\eta}$ such that $\boldsymbol{\eta}^T \boldsymbol{\eta} = \mathbf{I}_{\mathrm{d}}$, where $L(\boldsymbol{\eta})$ is proportional to a likelihood function, n_y is the number of observations in $\mathbb{X}$ with label y, and $|\cdot|$ denotes the determinant. This optimization is done through conjugate gradient descent on $\mathcal{G}(p, d)$. Details are given in [57]. In all experiments we use the `sgmin` implementation provided by Lippert and Edelman [72], which requires a closed-form first derivative and numerical second derivative.

5.3 Methodology

5.3.1 Regularized LAD

We propose a modified approach to LAD [73]. In order to obtain a useful model, we seek $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$ such that $[Y, \boldsymbol{\eta}^T \mathbf{X}] \sim [\Xi, \boldsymbol{\gamma}^T \mathbf{Z}]$. The objective function in (5.1) corresponds to the joint distribution for the reduced data from the source (i.e., $[\boldsymbol{\eta}^T \mathbf{X}]$). Moreover, noting that $[Y \,|\, \boldsymbol{\eta}^T \mathbf{X}] \sim [Y \,|\, \mathbf{X}]$ due to the sufficient dimension reduction approach, we see

$$[Y, \boldsymbol{\eta}^T \mathbf{X}] \sim [Y \,|\, \boldsymbol{\eta}^T \mathbf{X}][\boldsymbol{\eta}^T \mathbf{X}] \sim [Y \,|\, \mathbf{X}][\boldsymbol{\eta}^T \mathbf{X}]$$

and

$$[\Xi, \boldsymbol{\gamma}^T \mathbf{Z}] \sim [\Xi | \, \boldsymbol{\gamma}^T \mathbf{Z}][\boldsymbol{\gamma}^T \mathbf{Z}] \sim [\Xi | \, \mathbf{Z}][\boldsymbol{\gamma}^T \mathbf{Z}].$$

Recalling our assumption that $[Y | \, \mathbf{X}] \sim [\Xi | \, \mathbf{Z}]$, our goal now simply becomes to enforce $[\boldsymbol{\eta}^T \mathbf{X}] \sim [\boldsymbol{\gamma}^T \mathbf{Z}]$. Applying our assumption of within-class normality, this means we require

$$\boldsymbol{\eta}^T \boldsymbol{\mu}_{\mathrm{y}}^{\mathrm{x}} = \boldsymbol{\gamma}^T \boldsymbol{\mu}_{\mathrm{y}}^{\mathrm{z}}, \;\; \boldsymbol{\eta}^T \boldsymbol{\Sigma}_{\mathrm{y}}^{\mathrm{x}} \boldsymbol{\eta} = \boldsymbol{\gamma}^T \boldsymbol{\Sigma}_{\mathrm{y}}^{\mathrm{z}} \boldsymbol{\gamma}$$

for $y = 1, \ldots, C$. Since we will often not have adequate data from $\mathbf{Z}$, we forgo constraining the second moments. In cases where a large amount of target data is available a second constraint might be useful, though in practice we have found it results in poor performance. To constrain the first moments, we define a regularization term as

$$\Gamma_\lambda(\boldsymbol{\eta}, \boldsymbol{\gamma}) = \frac{\lambda}{2} \sum_{y=1}^{C} \left[\mathbf{1}\{m_y \geq 1\} || \, \boldsymbol{\eta}^T \boldsymbol{\mu}_{\mathrm{y}}^{\mathrm{x}} - \boldsymbol{\gamma}^T \boldsymbol{\mu}_{\mathrm{y}}^{\mathrm{z}} \, ||^2 \right]$$

for a fixed $\lambda > 0$. Here m_y is the number of observations from $\mathbb{Z}$ with response variable equal to y. We incorporate this regularization into the penalized likelihood

$$L^*(\boldsymbol{\eta}, \boldsymbol{\gamma}; \lambda) = L(\boldsymbol{\eta}; \mathbb{X}, \mathbf{y}) + L(\boldsymbol{\gamma}; \mathbb{Z}, \boldsymbol{\xi}) + \Gamma_\lambda(\boldsymbol{\eta}, \boldsymbol{\gamma})$$

and note that the gradient of the above likelihood with respect to $\boldsymbol{\eta}$ is proportional to

$$
\begin{aligned}
L^*_{\boldsymbol{\eta}}(\boldsymbol{\eta}, \boldsymbol{\gamma}; \lambda) &= \boldsymbol{\Sigma}^{\mathrm{x}}\, \boldsymbol{\eta} (\boldsymbol{\eta}^T\, \boldsymbol{\Sigma}^{\mathrm{x}}\, \boldsymbol{\eta})^{-1} - \sum_{y=1}^{C} \frac{n_y}{n}\, \boldsymbol{\Sigma}^{\mathrm{x}}_{\mathrm{y}}\, \boldsymbol{\eta} (\boldsymbol{\eta}^T\, \boldsymbol{\Sigma}^{\mathrm{x}}_{\mathrm{y}}\, \boldsymbol{\eta})^{-1} \\
&\quad + \lambda \sum_{y=1}^{C} \left[\mathbf{1}\{m_y \geq 1\} (\boldsymbol{\mu}^{\mathrm{x}}_{\mathrm{y}}\, \boldsymbol{\mu}^{\mathrm{xT}}_{\mathrm{y}}\, \boldsymbol{\eta} - \boldsymbol{\mu}^{\mathrm{x}}_{\mathrm{y}}\, \boldsymbol{\mu}^{\mathrm{zT}}_{\mathrm{y}}\, \boldsymbol{\gamma}) \right]
\end{aligned}
$$

and with respect to $\boldsymbol{\gamma}$ is

$$
\begin{aligned}
L^*_{\boldsymbol{\gamma}}(\boldsymbol{\eta}, \boldsymbol{\gamma}; \lambda) &= \boldsymbol{\Sigma}^{\mathrm{z}}\, \boldsymbol{\gamma} (\boldsymbol{\gamma}^T\, \boldsymbol{\Sigma}^{\mathrm{z}}\, \boldsymbol{\gamma})^{-1} - \sum_{y=1}^{C} \frac{m_y}{m}\, \boldsymbol{\Sigma}^{\mathrm{z}}_{\mathrm{y}}\, \boldsymbol{\gamma} (\boldsymbol{\gamma}^T\, \boldsymbol{\Sigma}^{\mathrm{z}}_{\mathrm{y}}\, \boldsymbol{\gamma})^{-1} \\
&\quad + \lambda \sum_{y=1}^{C} \left[\mathbf{1}\{m_y \geq 1\} (\boldsymbol{\mu}^{\mathrm{z}}_{\mathrm{y}}\, \boldsymbol{\mu}^{\mathrm{zT}}_{\mathrm{y}}\, \boldsymbol{\gamma} - \boldsymbol{\mu}^{\mathrm{z}}_{\mathrm{y}}\, \boldsymbol{\mu}^{\mathrm{xT}}_{\mathrm{y}}\, \boldsymbol{\eta}) \right].
\end{aligned}
$$

The benefit to this approach is that most computation will come from inverting the matrices $\boldsymbol{\eta}^T\, \boldsymbol{\Sigma}^{\mathrm{x}}_{\mathrm{y}}\, \boldsymbol{\eta}$ and $\boldsymbol{\gamma}^T\, \boldsymbol{\Sigma}^{\mathrm{z}}_{\mathrm{y}}\, \boldsymbol{\gamma}$. This means that, given adequate labeled data, we mitigate singularity issues when inverting this $d \times d$ matrix as opposed to, e.g., SIR, which requires an inverse of a $p \times p$ matrix. If d is relatively small and we have labeled data from the target space, we can estimate inverses in the above approach with minimal regularization of these matrices.

A potential drawback to the proposed approach is the estimation of two parameters ($\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$) as opposed to simply estimating one paramter $\boldsymbol{\eta}$ seen in many other approaches (cf. [65] for the $p = q$ case), though formulating the problem in this fashion will allow us to naturally handle cases in which $p \neq q$. For the case in which $p = q$, the same $\boldsymbol{\eta}$ is used for both training and testing data. We also seek param-

eters $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$ so that $[\boldsymbol{\eta}^T \mathbf{X}] \sim [\boldsymbol{\gamma}^T \mathbf{Z}]$, $[Y | \boldsymbol{\eta}^T \mathbf{X}] \sim [Y | \mathbf{X}]$, and $[\Xi | \boldsymbol{\gamma}^T \mathbf{Z}] \sim [\Xi | \mathbf{Z}]$. It is not explored in this chapter, but incorporating the conditional models directly into the objective could improve results.

For cases with nonlinearity, we localize the above method for a given observation $\mathbf{z}_0$ and bandwidth h by replacing $\boldsymbol{\mu}^{\mathrm{x}}$ and $\boldsymbol{\Sigma}^{\mathrm{x}}$ with the weighted mean and covariance matrix $\boldsymbol{\mu}^{\mathrm{x,h}}$ and $\boldsymbol{\Sigma}^{\mathrm{x,h}}$ where each observation is weighted by a kernel $K(\boldsymbol{\eta}^T \mathbf{x}_{\mathrm{i}}, \boldsymbol{\gamma}^T \mathbf{z}_0)$ for $i = 1, \ldots, n$. We use the initial estimates for $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$ to calculate $\boldsymbol{\mu}^{\mathrm{x,h}}$ and $\boldsymbol{\Sigma}^{\mathrm{x,h}}$ and perform a similar weighting in calculating the statistics from $[\Xi, \mathbf{Z}]$.

5.3.2 Grassmannian Data

Often cases arise in which data have an assumed prior structure, such as lying on the Grassmannian $\mathcal{G}(r, s)$, as has been seen in the previous chapters. An unfortunate consequence of this modification to the feature points is that the above dimension reduction approaches are not immediately applicable. While the predictors in this case can be concatenated into vectors and treated as Euclidean data, the reduced data will no longer have any relationship with the original manifold $\mathcal{G}(r, 2)$. Moreover, notions of the mean and the covariance of a set of points on a manifold are not as straightforward as their Euclidean counterparts.

We solve this problem by applying the inverse exponential map as described in Chapter 1, that is,

$$\exp^{-1}(\cdot;\boldsymbol{\mu}) : \mathcal{G}(r,2) \to \mathbb{R}^{2(r-2)}$$

defined on the Grassmannian that allows mapping between $\mathcal{G}(r,2)$ at a specified point $\boldsymbol{\mu} \in \mathcal{G}(r,2)$ and the corresponding tangent space. Since the tangent space about a given point on $\mathcal{G}(r,s)$ is simply $\mathbb{R}^{s(r-s)}$, we can now apply the above dimension reduction methods to the transformed data lying in this Euclidean space. In all experiments, we compute these mappings using numerical methods given in [10].

We are now left with the choice of $\boldsymbol{\mu}$. Typically this parameter is chosen to be an analogue to the mean of the given points, though since our observations $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are no longer Euclidean straightforward addition does not apply. Instead, as in Chapter 3 the "Fréchet mean" is defined as the point $\hat{\boldsymbol{\mu}}$ that satisfies (provided it exists)

$$\hat{\boldsymbol{\mu}} = \arg\min_{\boldsymbol{\mu}} \frac{1}{n} \sum_{i=1}^{n} \delta^2(\mathbf{x}_\mathrm{i}, \boldsymbol{\mu})$$

where δ^2 is a distance function defined on $\mathcal{G}(r,2)$. Since in most examples considered our data lie in a concentrated subset of the Grassmannian, we compute the sample mean and perform orthogonalization (e.g., through using singular value decomposition or the Gram-Schmidt procedure [67]) as it greatly reduced computation. See Chapter 3 for more detailed results.

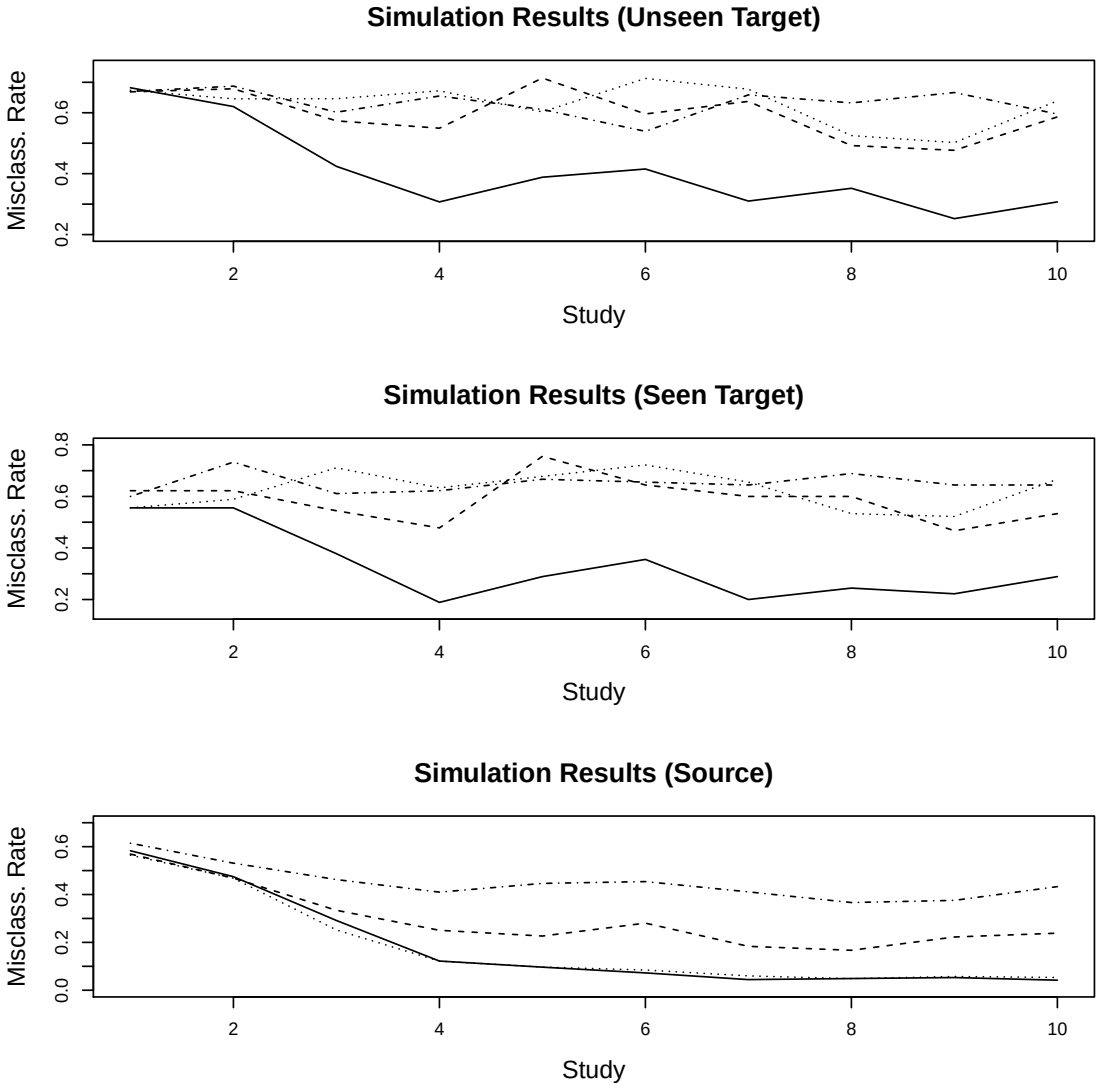


Fig. 5.2: Simulation study results with SIR (dashed line), LAD (dotted line), IS (dash-dot line), and RLD (solid line). Study j corresponds to $\alpha = 5/j^2$. For IS, 8 subspaces are used. For RLD, $\lambda = 1$ was used. For all methods, $d = 2$.

5.4 Simulation Studies

In all simulations, we test the proposed method (called RLD for "regularized likelihood directions," solid line) against various alternatives described above: SIR (dashed line), LAD (dotted line), and IS (dash-dot line). The latter method is modified to handle cases in which the ambient dimensions of each domain differ; we use

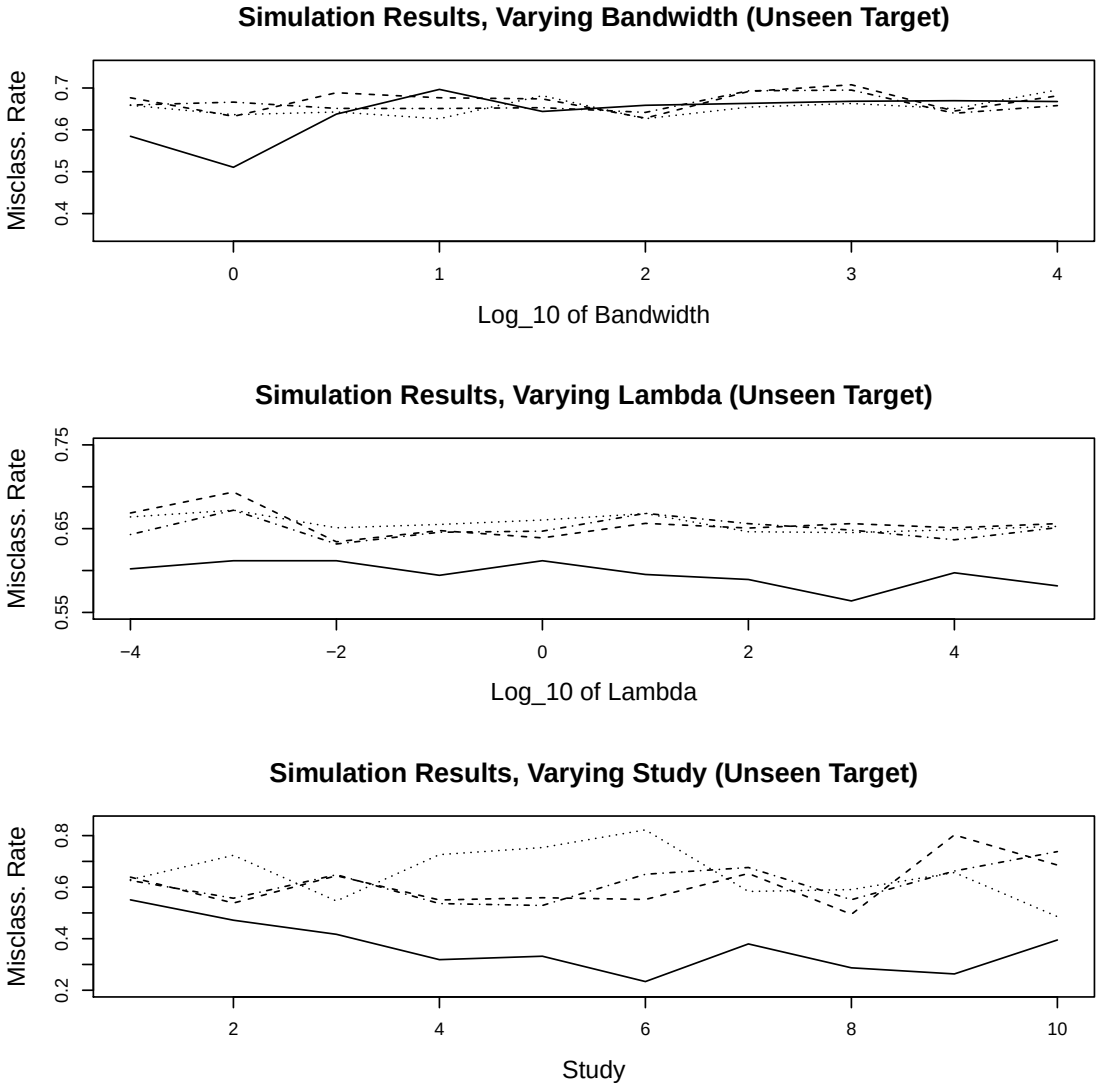


Fig. 5.3: Simulation study results with localization with SIR (dashed line), LAD (dotted line), IS (dash-dot line), and RLD (solid line). Top plot varies $h = 10^{(j-2)/2}$ for $j = 1, \ldots, 10$, $\lambda = 100$, and $\alpha = 5$. Middle plot varies $\lambda = 10^{j-5}$ for $j = 1, \ldots, 10$, $h = 10^{-.5}$, and $\alpha = 5$. Bottom plot varies the studies for $h = 1$ and $\lambda = 100$. All results are on unseen target data.

PCA to reduce the dimension of the higher-dimensional domain to be equal to that of the lower-dimensional domain. We assume a small number of labeled instances are in the target dataset, so we use the semisupervised extension of this method described in [52]. In all experiments, we use 8 intermediate subspaces. Due to the

singularity of $\mathbf{\Sigma}^{\mathrm{x}}$, we modify the SIR method to only take the top d eigenvectors of the matrix $\mathbf{M}\,\mathbf{M}^T$. In experiments, ignoring the inverse covariance matrix in SIR yields better results than any attempts at regularization or pseudoinversion. For LAD and RLD, the initial value for both $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$ is taken to be this SIR estimate. For SIR, LAD, and RLD we use multinomial logistic regression to obtain class labels for both the source and target domains. For IS, we use the PLS method with d latent directions in a one-vs-all classifier as oultined in [52] to estimate class labels in source and target.

For the simulation studies, we generate 200 observations in $\mathbb{R}^6$ for the source data and 300 observations in $\mathbb{R}^4$ for the target data to be tested (called the "unseen target data") in three classes. For the target data to be used in estimating $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$, we generate three observations per class distributed the same as the unseen target data, called the "seen target data." We generate $\mathbf{X}$ and $\mathbf{Z}$ as in Chapter 2 and generate both $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$ with random normal entries in $\mathbb{R}^{6\times 2}$ and $\mathbb{R}^{4\times 2}$, respectively, and take their orthogonalizations. The labels are generated as

$$y = \boldsymbol{\eta}^T \mathbf{x}\,/\alpha - \alpha \cdot (\boldsymbol{\eta}^T \mathbf{x})^2 + \epsilon$$

where $\epsilon \sim N(0, .5^2)$ and α is a chosen parameter that governs the weight placed on the linear term in the model. The response values y are then discretized into three categories by thresholding them at their one-third and two-thirds quantiles. Similar labels are generated for the target data using $\boldsymbol{\gamma}$.

Each study is run ten times, and the average misclassification rate is recorded

for all methods under consideration. Results are given in Fig. 5.2. For study j, we take $\alpha = 5/j^2$ so as the study number increases, less weight is given to the quadratic term in the model. For the target data, the RLD method has clear advantages when the underlying model generating the labels is linear, though it seems to lose some of its predictive power for models that exhibit some nonlinearity. LAD and RLD yielded similar results when applied to the source data.

In an attempt to improve performance for the models with higher weight on the quadratic term, we perform localization using a radial Gaussian kernel with bandwidth $h = 1$ for each method. Fig. 5.3 shows the results of the localized dimension reduction on just the unseen target data. The top plot shows that low values for the bandwidth yield better results for RLD, while the competing methods do not show any measured improvements as the bandwidth increases. As we vary λ in the middle plot, we do not see much improvement in performance for RLD. Due to this we do not vary λ much in the applications to real data. In the bottom plot, we see that RLD outperforms the alternatives in every study, and for all methods we see some improvement for the more quadratic models.

5.5 Case Studies

In addition to the methods used for comparison in the simulation studies, we also compare with two alternative domain adaptation algorithms on real-world data. We use metric learning (MLR, [4]) with one-nearest-neighbor classification and $\gamma = 100$ as given in the sample code provided by the authors. In order to compare

Table 5.1: Object recognition results, source: HOG features, target: HOG features (top), raw image data (bottom). For RLD, $\lambda = .2$. For IS, we use 8 subspaces. For MLR, $\gamma = 100$. For all methods, $d = 10$. All results are on unseen target data. Here A:W denotes `amazon.com` source and webcam as target, A:D denotes `amazon.com` source and DSLR target, etc.

Same Dimension						
	A:W	**A:D**	**W:A**	**W:D**	**D:A**	**D:W**
SIR	10.22 (1.93)	13.23 (2.23)	8.99 (1.23)	12.65 (3.17)	8.04 (2.67)	12.83 (4.28)
LAD	11.80 (2.48)	11.06 (2.63)	8.24 (1.38)	14.10 (4.57)	6.87 (2.05)	13.12 (3.26)
IS	15.12 (1.52)	14.51 (2.16)	11.34 (0.97)	27.17 (1.38)	**12.48** (0.60)	21.97 (1.81)
GFK	10.85 (1.14)	10.80 (1.52)	11.45 (0.45)	**35.88** (1.80)	12.21 (1.14)	**36.85** (2.06)
MLR	9.74 (2.38)	8.08 (1.78)	8.58 (0.99)	20.62 (3.14)	9.28 (1.90)	27.01 (3.04)
RLD	**15.89** (2.19)	**19.70** (2.41)	**11.88** (1.64)	26.19 (2.54)	10.73 (1.36)	24.47 (2.77)
Different Dimension						
SIR	4.54 (1.66)	4.00 (0.93)	3.89 (1.12)	3.23 (2.48)	3.87 (1.06)	3.31 (1.91)
LAD	3.79 (1.36)	3.87 (1.68)	3.72 (1.24)	3.94 (2.03)	4.46 (0.65)	4.19 (1.48)
IS	2.87 (1.32)	2.76 (1.53)	2.74 (0.83)	2.14 (1.07)	3.04 (0.64)	2.61 (1.61)
GFK	3.58 (1.35)	2.80 (0.97)	3.18 (1.06)	3.31 (1.14)	2.67 (0.84)	3.98 (1.17)
MLR	3.30 (1.19)	3.04 (1.08)	3.22 (1.19)	2.88 (1.37)	3.54 (0.76)	2.78 (1.67)
RLD	**16.70** (1.78)	**17.31** (4.12)	**11.59** (2.65)	**18.38** (2.04)	**11.38** (2.32)	**13.91** (1.50)

with the dimension reduction methods given, we use a rank-d approximation to the kernel estimated by this method. We also use GFK with one-nearest-neighbor and d dimensions as given in the sample code provided by the authors. For the case in which the source and target domain have differing dimensions, we modify the methods as we did for IS in the simulation studies.

5.5.1 Euclidean Data

We focus on the object recognition dataset from [4] and Chapter 4. In it, three domains are given: data collected from `amazon.com`, data collected from a high-resolution DSLR camera, and data collected from a low-resolution web-cam. Images from [4] are also converted to grayscale and downsampled to 10×10 images. See Fig. 5.1 for example images.

5.5.1.1 Object Recognition

We conduct two different experiments, each consisting of six separate studies corresponding to each combination of souce and target, e.g., the first study uses the `amazon.com` data as source and the webcam data as target (A:W), the second study uses the `amazon.com` data as source and the DSLR data as target (A:D), etc. In all studies, we randomly sample 20 observations per class from the source data with replacement and 3 observations per class from the target data with replacement, using these observations for estimating $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$ (the "source" and "seen target" data, respectively). We then test on the target data that was not used in estimating these parameters. This is done ten times, and we record the average and standard deviation of the classification rate.

In the first experiment, we use the HOG features of Chapter 4 for both the source and target data (denoted "same dimension"). The results from this experiment are given at the top of Table 5.1. We do not use localization, as the improvements are only slight in the simulation studies with a large increase in computation. When testing on the unseen target data, RLD outperforms the other methods in the first three studies and is competitive for DSLR camera as source and `amazon.com` as target. GFK performs much better than the alternatives when there is not a large discrepancy between source and target (i.e., cases D:W and W:D).

In the second experiment, the HOG features from the first experiment are used for the source data, whereas for the target data each image is read as grayscale values between zero and one and resized to 10×10 pixels, then concatenated into a vector in

$\mathbb{R}^{100}$ (denoted "different dimension"). Results are given at the bottom of Table 5.1. For this experiment, on the unseen target data we see large drops in performance for all methods except RLD, which drops only slightly and now outperforms all competing methods in all studies. For this experiment, the likelihood-acquired directions (LAD) method seems to perform next best, though all alternative methods show performances on par with random guessing.

5.5.2 Grassmannian Data

For data with a known structure, we focus on the landmark points given in FG-NET.

5.5.2.1 Age Estimation

For age estimation, we generate three different target domains: the first rotates each image by an angle sampled uniformly at random from zero to $\pi/4$; for the second and third target datasets, we remove 17 and 34 landmark points at random from this rotated dataset. The square-root of the age variable is used as a response variable as it results in positive estimates for age and has been shown to work better in practice [67]. The estimated dimension for study i is $\lfloor (q/2-2)/i \rfloor$ where $q = 136, 102, 68$ and i ranges from one to five.

The approach to estimating $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$ for RLD in the case of continuous response is the same as for discrete response, but follows the binning method common in the inverse regression literature [71]. For experiments, we categorize the continuous age

Table 5.2: Age estimation results, source: full landmark points, target: full landmark points (top), three-fourths landmark points (middle), one-half landmark points (bottom). For RLD, $\lambda = 4$. For IS, we use 8 subspaces. For MLR, $\gamma = 100$. All results are on unseen target data.

		i = 1	**i = 2**	**i = 3**	**i = 4**	**i = 5**
r = 68	SIR	9.51 (2.00)	10.32 (2.27)	8.78 (1.91)	9.40 (2.66)	9.01 (2.28)
	LAD	10.12 (1.86)	9.52 (1.63)	8.23 (1.25)	8.27 (1.05)	8.09 (1.03)
	IS	**6.94** (0.09)	6.76 (0.24)	**6.31** (0.09)	6.30 (0.09)	6.60 (0.21)
	GFK	7.09 (0.48)	**6.60** (0.29)	6.41 (0.35)	**6.28** (0.25)	**6.24** (0.23)
	MLR	19.41 (4.87)	19.37 (7.17)	21.62 (7.11)	16.76 (3.91)	19.19 (4.39)
	RLD	7.85 (0.37)	7.72 (0.50)	7.61 (0.40)	7.46 (0.41)	7.83 (0.59)
r = 51	SIR	10.57 (1.70)	9.02 (2.05)	8.87 (1.51)	8.21 (0.59)	10.21 (3.07)
	LAD	10.19 (1.45)	9.16 (2.00)	8.73 (1.04)	8.23 (0.48)	8.33 (1.33)
	IS	9.91 (0.87)	9.49 (0.52)	9.66 (1.18)	8.74 (0.72)	9.15 (0.79)
	GFK	12.38 (1.43)	14.00 (2.18)	13.85 (2.32)	13.77 (1.93)	12.34 (1.07)
	MLR	14.37 (2.52)	15.91 (5.72)	15.40 (2.91)	16.12 (4.34)	15.84 (4.78)
	RLD	**8.30** (0.82)	**7.59** (0.26)	**7.84** (0.53)	**7.62** (0.39)	**7.66** (0.31)
r = 34	SIR	**9.92** (1.32)	10.80 (1.67)	9.60 (1.79)	10.16 (2.10)	9.66 (2.49)
	LAD	10.26 (1.32)	10.76 (1.59)	9.49 (1.61)	9.28 (1.86)	9.02 (1.98)
	IS	10.16 (0.53)	9.85 (0.77)	9.51 (0.81)	9.74 (0.71)	10.06 (0.57)
	GFK	13.04 (0.72)	13.30 (0.95)	12.66 (0.90)	12.62 (0.96)	12.24 (0.97)
	MLR	16.10 (3.73)	15.90 (4.88)	14.68 (3.19)	17.13 (5.49)	15.70 (3.28)
	RLD	10.66 (1.63)	**9.32** (1.13)	**8.71** (0.45)	**8.12** (0.48)	**8.10** (0.53)

variable into two categories. Increasing the number of categories does not seem to make a large difference in prediction, though it does increase computation. For source data, 250 observations are randomly sampled from each category, and for seen target data 3 observations are sampled. Table 5.2 shows mean absolute errors for each of the different target domains. Each study is run on various estimates for the reduced dimension in the tangent space. As in the object recognition experiments, each of these studies is run ten times and averaged, here reporting the average and standard deviation of the mean absolute errors.

We see that both the IS and GFK method perform well in the case in which the source and target domains are of the same dimension, though RLD is somewhat competitive. When the dimension of the target domain differs from that of the source, RLD outperforms all methods, with the exception of one case in which

Table 5.3: Face recognition results, source: full landmark points, target: full landmark points (top), three-fourths landmark points (middle), one-half landmark points (bottom). For RLD, $\lambda = 4$. For IS, we use 8 subspaces. For MLR, $\gamma = 100$. All results are on unseen target data.

		i = 1	**i = 2**	**i = 3**	**i = 4**	**i = 5**
r = 68	SIR	4.65 (1.20)	6.00 (1.30)	6.14 (2.04)	6.66 (1.63)	6.77 (1.54)
	LAD	4.99 (1.01)	5.82 (1.87)	6.70 (1.70)	6.15 (2.46)	6.47 (1.37)
	IS	**14.35** (3.02)	**23.54** (3.47)	**28.11** (2.50)	**29.06** (2.38)	**29.93** (3.64)
	GFK	4.30 (1.07)	4.04 (0.93)	4.96 (0.97)	4.05 (1.50)	5.50 (1.28)
	MLR	5.42 (2.07)	9.36 (1.66)	12.12 (3.06)	13.34 (1.32)	15.38 (2.43)
	RLD	9.51 (1.84)	12.43 (1.93)	13.01 (2.90)	13.36 (2.52)	14.06 (3.74)
r = 51	SIR	4.68 (0.69)	5.90 (1.04)	6.05 (1.29)	6.89 (1.54)	7.66 (1.25)
	LAD	4.90 (1.53)	5.77 (1.25)	5.96 (1.33)	6.72 (1.02)	7.57 (1.64)
	IS	3.22 (0.83)	3.01 (1.23)	3.90 (1.32)	7.10 (1.63)	7.52 (2.54)
	GFK	1.85 (0.95)	1.55 (0.86)	1.50 (0.79)	1.25 (0.72)	1.24 (0.79)
	MLR	1.55 (0.95)	2.58 (0.97)	4.89 (1.26)	6.05 (1.80)	8.42 (1.54)
	RLD	**8.93** (1.86)	**11.58** (2.07)	**11.29** (3.17)	**12.14** (2.20)	**13.50** (1.47)
r = 34	SIR	3.91 (1.11)	4.38 (1.19)	6.39 (1.35)	6.79 (1.19)	6.20 (1.63)
	LAD	3.30 (0.95)	4.60 (0.99)	5.96 (1.23)	5.33 (1.19)	6.11 (1.21)
	IS	2.31 (0.70)	2.58 (1.41)	3.86 (1.72)	4.38 (1.21)	4.86 (1.81)
	GFK	1.55 (0.73)	1.16 (0.61)	1.46 (0.71)	1.76 (1.47)	1.29 (0.67)
	MLR	1.59 (0.84)	1.98 (0.76)	3.86 (1.38)	3.60 (1.70)	5.53 (1.34)
	RLD	**5.71** (1.62)	**8.38** (1.31)	**12.14** (0.81)	**12.20** (1.46)	**12.74** (2.40)

the estimated dimension is small. Note that RLD seems to be fairly stable in its predictive performance regardless of the dimensions involved, so that while a small price is paid in the case of similar dimensions, improvements can be made when there is less information in the target space.

5.5.2.2 Face Recognition Across Aging

For face recognition across aging, we split the observations into two domains: individuals younger than 19 and individuals 19 and older. The goal is to perform recognition on the individuals using under-19 data as source while testing on the 19-and-over data. We remove all individuals who do not have at least one observation from each of these domains, resulting in 724 total observations comprising 60 individuals to classify. We do not rotate images first as the difference in age groups is meant to be a large enough domain shift in this case. For the second and third

target datasets we again remove 17 and 34 landmark points at random. Because of the small number of observations within-class, we randomly sample 3 observations per class from the source domain and 1 observation per class from the target to train the model, while testing on all remaining target data, with recognition rates and corresponding standard deviations given in Table 5.3.

As in the case of age estimation, IS performs well in the same-dimension case with RLD being only slightly competitive, GFK performing much worse. For cases in which the dimensions differ, we see RLD outperforms all competing methods handily. Due to the small sample sizes within-class, we run into issues with estimation and identifiability of the within-class statistics. Shrinkage methods (e.g., [32]) do not seem to yield any improvements in the results, though these methods might help matters when including additional regularization terms.

5.6 Extension: Incorporating Transformations

A benefit to obtaining a single dimension reduction parameter is that we are afforded with the possibility of exploratory analysis using graphical methods. For example, in the simulation study above we have a quadratic term in the true model. Many methods, such as GFK or MLR, have no way of discovering this information, nor any way to incorporate it into a final model, and it is similarly unclear how to extend IS to take this model into account.

We run a single instance of the simulation for a continuous y and $\alpha = 5$. We simulate 200 observations from the source distribution and 30 from the target to

be used in obtaining the dimension reduction parameters. We plot the directions obtained versus the response in Fig. 5.4. We see clear nonlinearity in the directions obtained through RLD, though unfortunately the nonlinearity is not necessarily exhibited in the same directions. We can use this information to improve our model fit, and see in fact that the MAE for the 300 unseen target points using the linear terms is 12.306 while including all quadratic terms drops it to 8.809. This approach could possibly be used to improve prediction as well as interpretability in cases where data inhomogeneity occurs. This discrepancy between the directions in which structure is exhibited could indicate that alternative penalty terms could be incorporated into the likelihood for $\boldsymbol{\eta}$, potentially involving the conditional model $[Y \,|\, \mathbf{X}]$.

5.7 Discussion

We have shown that, by adding a regularization term to the likelihood function of likelihood-acquired directions we are able to yield improvements over simply using the directions themselves in cases in which the distribution of the predictors changes from training to testing. Additionally, these directions produce a linear transformation that both reduces dimension and can adequately discriminate between categories in a variety of classification tasks. RLD outperforms SIR and LAD, two related dimension reduction methods, and it outperforms IS, MLR, and GFK – all methods that similarly try to take into account the discrepancy between training and testing distributions – when the dimensions of the training and testing data differ. RLD can extend to cases where predictors have a prior structure, and

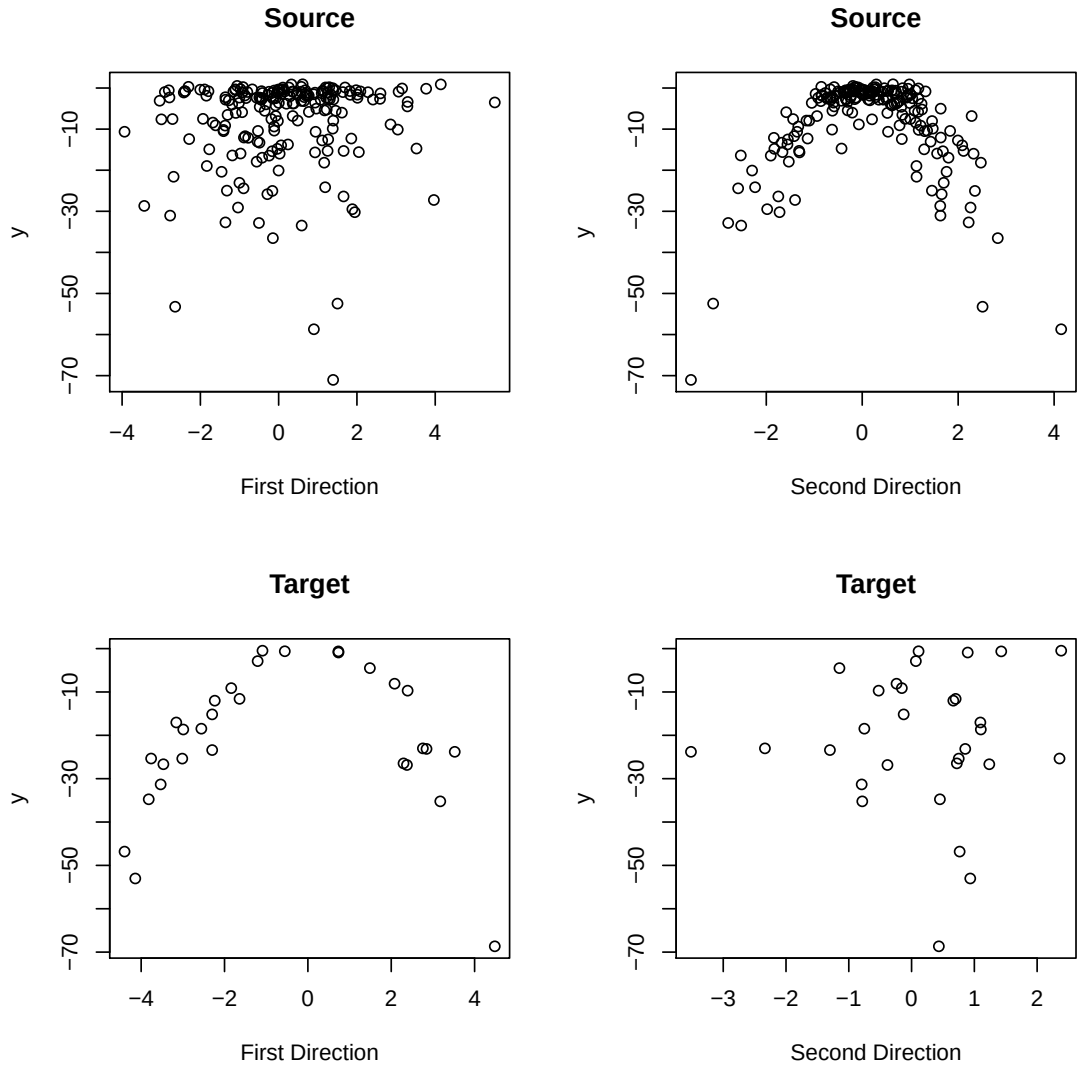


Fig. 5.4: Plots of the source and target data directions found through RLD.

localization is straightforward and can improve performance when the underlying model is nonlinear.

6

Monte Carlo Acquired Directions - Preliminary Results

6.1 Introduction

We have seen that both CDE and RLD can yield gains in predictive performance when data are inhomogeneous between training and testing; CDE is able to incorporate information from the response in constructing a dimension reduction, though in this case it is done through a penalized maximum likelihood framework on the response given the reduced predictor. RLD takes the tack from sufficient dimension reduction literature and maximizes the likelihood of the response and the predictors over the transformation directly while trying to handle inhomogeneous data through regularization. This regularization attempts to keep within-class first moments of the source and target distribution close, though ideally we would like to have both first and second moments close to one another. Most penalty terms for this latter constraint are not ideal: they yield gradients in which the desired transformation appears in a third-order term, often resulting in poor convergence. We propose a different approach to incorporate both distributional constraints as well as include information from the conditional model in construction dimension reductions.

6.2 Methodology

Our approach will be as in Chapter 5 in which we assume a normal model for $[\mathbf{X}\,|Y]$ and $[\mathbf{Z}\,|\Xi]$ and assume independent predictors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and $\mathbf{z}_1, \ldots, \mathbf{z}_m$ with corresponding response values. Previously, we used penalized maximum likelihood to estimate dimension reduction parameters $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$ that would be good "sufficient" dimension reductions while constraining the distributions of $[\boldsymbol{\eta}^T \mathbf{X}\,|Y]$ and $[\boldsymbol{\gamma}^T \mathbf{Z}\,|\Xi]$ to be close. In our previous method we used only the first moments of these distributions in the constraint as higher moment constraints complicated the gradient and did not yield useful results in practice. In the current approach, we will use sequential Monte Carlo sampling [74] to avoid computing a gradient of a quartic function of the parameter while still attempting to incorporate second moment constraints. We will also incorporate the conditional model $[Y|\,\mathbf{X}, \boldsymbol{\eta}, \boldsymbol{\beta}]$ in the construction of the dimension reduction parameters. In RLD, the full likelihood can be written as

$$\pi(\boldsymbol{\eta}, \boldsymbol{\gamma}) \sim [Y|\,\boldsymbol{\eta}^T \mathbf{x}, \boldsymbol{\beta}] \cdot [\Xi|\,\boldsymbol{\gamma}^T \mathbf{z}, \boldsymbol{\beta}] \cdot [\boldsymbol{\eta}^T \mathbf{x}] \cdot [\boldsymbol{\gamma}^T \mathbf{z}] \cdot [\boldsymbol{\eta}, \boldsymbol{\gamma}] \tag{6.1}$$

where $[\boldsymbol{\eta}, \boldsymbol{\gamma}]$ is the term to constrain the within-class means of $\mathbf{x}$ and $\mathbf{z}$ to be similar. We note that, though (6.1) depends on a parameter $\boldsymbol{\beta}$, we will take this value to be fixed given $\{\boldsymbol{\eta}, \boldsymbol{\gamma}\}$. The current approach will be to proceed similar to RLD, but with an eye on the covariance structure of each domain; in other words, we will modify the term $[\boldsymbol{\eta}, \boldsymbol{\gamma}]$ to incorporate second moment constraints. As our data is assumed to be normal within-class, constraining $\{\boldsymbol{\eta}^T \boldsymbol{\mu}_{\mathrm{y}}^{\mathrm{x}}, \boldsymbol{\eta}^T \boldsymbol{\Sigma}_{\mathrm{y}}^{\mathrm{x}} \boldsymbol{\eta}\}$ to be equivalent

to $\{\boldsymbol{\gamma}^T \boldsymbol{\mu}_{\mathrm{y}}^{\mathrm{z}}, \boldsymbol{\gamma}^T \boldsymbol{\Sigma}_{\mathrm{y}}^{\mathrm{z}} \boldsymbol{\gamma}\}$ will be all that we require.

6.3 Choice of Prior

A sequential Monte Carlo algorithm will require prior distributions for $\boldsymbol{\eta}$, and $\boldsymbol{\gamma}$. One desirable property of such distributions will be that they are easy to sample from, as we will deem it necessary to draw a large number of variates from these distributions. As $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$ will lie on the Grassmannian $\mathcal{G}$ we require a bit more machinery for their prior distributions. A popular distribution for $\boldsymbol{\eta}$ is the "Procrustean" prior (see [2] and Chapter 3)

$$\pi_{0,\eta} \sim \mathrm{etr}\{-[\mathbf{I}_{\mathrm{d}} - \boldsymbol{\eta}^T \hat{\boldsymbol{\eta}}^{\mathrm{x}} (\hat{\boldsymbol{\eta}}^{\mathrm{x}})^T \boldsymbol{\eta}]/\sigma_\eta^2\} \tag{6.2}$$

where $\hat{\boldsymbol{\eta}}^{\mathrm{x}}$ is an initial estimate of $\boldsymbol{\eta}$ using $(\mathbf{y}, \mathbb{X})$. Here σ_η^2 is a chosen parameter. An unfortunate property of this distribution is that, in order to obtain variates from it, we require the use of a rejection sampler; that is, to simulate a variate from (6.2), we first generate a random uniform variate $\mathbf{U}$ on $\mathcal{G}(p, d)$ (see Chapter 3, Table 3.2), then generate a random uniform variate $u \sim U(0, 1)$, accepting $\mathbf{U}$ if $u < \pi_{0,\eta}(\mathbf{U})$ and rejecting $\mathbf{U}$ otherwise. This has the potential to be computationally intensive; if σ_η^2 is small enough we may generate a large number of proposals for $\mathbf{U}$ before accepting. For this reason, we consider the wrapped normal distribution [21]. We recall (1.1) and (1.2) from Chapter 1; let the geodesic between $\boldsymbol{\eta}_{\mathrm{j}}$ and $\boldsymbol{\eta}_{\mathrm{k}}$ be written as

$$\boldsymbol{\delta}(t;\boldsymbol{\eta}_{\mathrm{j}},\boldsymbol{\eta}_{\mathrm{k}}) = \boldsymbol{\eta}_{\mathrm{j}}\,\mathbf{U}_1\,\boldsymbol{\Gamma}(t) - \boldsymbol{\eta}_{\mathrm{j}}^{\perp}\,\mathbf{U}_2\,\boldsymbol{\Sigma}(t)$$

where $\mathbf{U}_1, \mathbf{U}_2, \boldsymbol{\Gamma}, \boldsymbol{\Sigma}$ are given by the generalized singular value decomposition

$$\boldsymbol{\eta}_{\mathrm{j}}^T\,\boldsymbol{\eta}_{\mathrm{k}} = \mathbf{U}_1\,\boldsymbol{\Gamma}\,\mathbf{V}^T,\ (\boldsymbol{\eta}_{\mathrm{j}}^{\perp})^T\,\boldsymbol{\eta}_{\mathrm{k}} = -\,\mathbf{U}_2\,\boldsymbol{\Sigma}\,\mathbf{V}^T\,. \tag{6.3}$$

Moreover, we recall from Chapter 1 that we have the maps about the point $\boldsymbol{\eta}_0$

$$\exp(\cdot,\boldsymbol{\eta}_0) : \mathbb{R}^{d(p-d)} \to \mathcal{G}(p,d),\ \exp^{-1}(\cdot,\boldsymbol{\eta}_0) : \mathcal{G}(p,d) \to \mathbb{R}^{d(p-d)}.$$

For a point $\mathbf{R} \in \mathcal{G}(p,d)$, we can take

$$\exp(\mathbf{R};\boldsymbol{\eta}_0) = \boldsymbol{\delta}(1;\boldsymbol{\eta}_0,\mathbf{R}),$$

which, using (6.3), shows that

$$\exp(\mathbf{R};\boldsymbol{\eta}_0) = \boldsymbol{\eta}_0\,\mathbf{U}_1\,\boldsymbol{\Gamma} - \boldsymbol{\eta}_0^{\perp}\,\mathbf{U}_2\,\boldsymbol{\Sigma} = \mathbf{U}\,\mathbf{V}(\mathbf{V}^T\,\mathbf{V})^{-1}$$

as

$$\mathbf{R} = [\boldsymbol{\eta}_0\ \ \boldsymbol{\eta}_0^{\perp}] \begin{bmatrix} \mathbf{U}_1\,\boldsymbol{\Gamma}\,\mathbf{V}^T \\ -\,\mathbf{U}_2\,\boldsymbol{\Sigma}\,\mathbf{V}^T \end{bmatrix}.$$

Now, we take

$$\mathbf{U} \sim N_{pd}(\text{vec}[\boldsymbol{\eta}_0], \sigma_\eta^2 \mathbf{I})$$

and for a block-diagonal matrix

$$\mathbf{Q} = \text{blockdiag}[(\mathbf{V}^T \mathbf{V})^{-1} \mathbf{V}^T]$$

we can reform $\mathbf{Q}\,\mathbf{U}$ column-wise into $\tilde{\boldsymbol{\eta}} \in \mathbb{R}^{p \times d}$ with $\tilde{\boldsymbol{\eta}}^T \tilde{\boldsymbol{\eta}} = \mathbf{I}_\text{d}$. We now have two benefits to the above formulation: first, we are able to generate $\mathbf{U}$ as multivariate normal and map it directly into $\mathcal{G}(p, d)$ about an initial point $\boldsymbol{\eta}_0$; second, as it will be required in our later Metropolis-Hastings algorithm, we have an approximation to the density function of a random variate $\tilde{\boldsymbol{\eta}}$, that is,

$$\text{vec}(\tilde{\boldsymbol{\eta}}) \sim N_{pd}(\text{vec}[\boldsymbol{\eta}_0], \sigma_\eta^2 \mathbf{Q}^T \mathbf{Q}) \tag{6.4}$$

where $\text{vec}(\tilde{\boldsymbol{\eta}})$ is the column-wise concatenation of $\tilde{\boldsymbol{\eta}}$ into a vector in $\mathbb{R}^{pd}$. We stress that this is only an approximation, as $\mathbf{Q}$ will be a function of $\tilde{\boldsymbol{\eta}}$.

6.4 Sequential Monte Carlo

In sequential Monte Carlo (SMC), at stage s we let

$$\pi_s \propto \pi_0^{1-\alpha_s} \pi^{\alpha_s},$$

$$\pi_0 \equiv \pi_{0,\eta} \cdot \pi_{0,\gamma},$$

Table 6.1: Summary of sequential Monte Carlo algorithm

- Initialize π_0 above for $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$.
- Initially sample $i = 1, \ldots, N$ particles $\boldsymbol{\eta}_i^0$ and $\boldsymbol{\gamma}_i^0$ from π_0 and set initial weights $w_i \equiv 1/N$.
- At stage s:
  - Set $w_i^s = w_i^{s-1}(\boldsymbol{\eta}_i^{s-1}, \boldsymbol{\gamma}_i^{s-1})/\pi_0(\boldsymbol{\eta}_i^{s-1}, \boldsymbol{\gamma}_i^{s-1}))^{\alpha_s - \alpha_{s-1}}$ and normalize.
  - If effective sample size [here $(\sum_i w_i)^2/\sum_i w_i^2$] is less than $N/2$, resample particles with replacement where element i is selected with probabiliy w_i^s, then reset weights to $1/N$.
  - Generate proposals $\mathrm{vec}(\tilde{\boldsymbol{\eta}}_i^{s-1}) \sim N_{pd}(\mathrm{vec}[\boldsymbol{\eta}_i^{s-1}], \sigma_\eta^2 \mathbf{Q}^T \mathbf{Q})$ accepted with probability
  $$\rho_i = \min\left\{1, \frac{\pi_s(\tilde{\boldsymbol{\eta}}_i^{s-1}, \boldsymbol{\gamma}_i^{s-1})}{\pi_s(\boldsymbol{\eta}_i^{s-1}, \boldsymbol{\gamma}_i^{s-1})}\right\},$$
  i.e., on acceptance, set $\boldsymbol{\eta}_i^s = \tilde{\boldsymbol{\eta}}_i^{s-1}$, otherwise set $\boldsymbol{\eta}_i^s = \boldsymbol{\eta}_i^{s-1}$.
  - Similar to $\tilde{\boldsymbol{\eta}}_i^{s-1}$, generate proposals $\tilde{\boldsymbol{\gamma}}_i^{s-1}$ from $\pi_{0,\gamma}$ using $\boldsymbol{\gamma}_i^{s-1}$.

$$0 = \alpha_0 \leq \alpha_1 \leq \ldots \leq \alpha_S = 1,$$

using (6.4) for $\pi_{0,\eta}$ and $\pi_{0,\gamma}$. The sampling strategy is given in Table 6.1.

6.4.1 Inhomogeneous Data Term

As we assume that our data is inhomogeneous, we add a term $\pi_{\eta,\gamma}$ to the posterior as an effective "prior" incorporating this inhomogeneity; this is much like the $[\boldsymbol{\eta}, \boldsymbol{\gamma}]$ term described earlier (e.g., the first moment constraint of Chapter 5). Since the parameters $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$ will be acquired through log determinants of the covariance matrices $\boldsymbol{\Sigma}^{\mathrm{x}}$ and $\boldsymbol{\Sigma}^{\mathrm{z}}$, as well as the within-class covariance matrices, we will consider the constraints

$$\text{C1}: \ \log \pi_{\eta,\gamma} \ = \ \frac{1}{\sigma^2_{\eta,\gamma}} \Bigg[(n+m) \cdot || \, \boldsymbol{\eta}^T \, \boldsymbol{\mu}^{\text{x}} - \boldsymbol{\gamma}^T \, \boldsymbol{\mu}^{\text{z}} \, ||^2 + \sum_y (n_y + m_y) \cdot || \, \boldsymbol{\eta}^T \, \boldsymbol{\mu}^{\text{x}}_{\text{y}} - \boldsymbol{\gamma}^T \, \boldsymbol{\mu}^{\text{z}}_{\text{y}} \, ||^2 \Bigg],$$

$$\text{C2}: \ \log \pi_{\eta,\gamma} \ = \ \frac{1}{\sigma^2_{\eta,\gamma}} \Bigg[(n+m) \cdot (\log | \, \boldsymbol{\eta}^T \, \boldsymbol{\Sigma}^{\text{x}} \, \boldsymbol{\eta} \, | - \log | \, \boldsymbol{\gamma}^T \, \boldsymbol{\Sigma}^{\text{z}} \, \boldsymbol{\gamma} \, |)^2 + \sum_y (n_y + m_y) \cdot (\log | \, \boldsymbol{\eta}^T \, \boldsymbol{\Sigma}^{\text{x}}_{\text{y}} \, \boldsymbol{\eta} \, | - \log | \, \boldsymbol{\gamma}^T \, \boldsymbol{\Sigma}^{\text{z}}_{\text{y}} \, \boldsymbol{\gamma} \, |)^2 \Bigg]$$

where C1 will be the same as the within-class mean constraint proposed in Chapter 5. The second constraint is considered as we wish for within-class second moments of the reduced data to be equivalent, though our objective depends on these moments through the $\log | \cdot |$ function.

6.4.2 Posterior

For this method, we require a density proportional to the posterior $\pi(\boldsymbol{\eta}, \boldsymbol{\gamma})$. Combining all of the above, we see the log of the posterior distribution will be

$$
\begin{aligned}
\log \pi(\boldsymbol{\eta}, \boldsymbol{\gamma}) \;\; \sim \;\; & -\frac{1}{2} \Bigg\{ \frac{1}{\sigma_{ex}^2} || \, \mathbf{y} - \mathbb{X} \, \boldsymbol{\eta} \, \tilde{\boldsymbol{\beta}} ||^2 + \frac{1}{\sigma_{ez}^2} || \, \boldsymbol{\xi} - \mathbb{Z} \, \boldsymbol{\gamma} \, \tilde{\boldsymbol{\beta}} ||^2 \\
& - \left[\log | \, \boldsymbol{\eta}^T \, \boldsymbol{\Sigma}^{\mathrm{x}} \, \boldsymbol{\eta} \, | - \sum_y \frac{n_y}{n} \log | \, \boldsymbol{\eta}^T \, \boldsymbol{\Sigma}_{\mathrm{y}}^{\mathrm{x}} \, \boldsymbol{\eta} \, | \right] \\
& - \left[\log | \, \boldsymbol{\gamma}^T \, \boldsymbol{\Sigma}^{\mathrm{z}} \, \boldsymbol{\gamma} \, | - \sum_y \frac{m_y}{m} \log | \, \boldsymbol{\gamma}^T \, \boldsymbol{\Sigma}_{\mathrm{y}}^{\mathrm{z}} \, \boldsymbol{\gamma} \, | \right] \\
& + \log \pi_{\eta,\gamma} \Bigg\} \qquad (6.5)
\end{aligned}
$$

where this posterior depends on both samples $(\mathbf{y}, \mathbb{X})$ and $(\boldsymbol{\xi}, \mathbb{Z})$, and $\tilde{\boldsymbol{\beta}}$ is estimated from the current values of $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$.

6.5 Preliminary Results

6.5.1 Simulation

For the simulation studies we proceed similarly to that Chapter 5. We generate 1000 observations in $\mathbb{R}^6$ for the source data and 1000 observations in $\mathbb{R}^4$ for the target data to be tested. For the target data to be used in estimating $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$, we generate 30 observations per class distributed. We generate $\mathbb{X}$ and $\mathbb{Z}$, $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$, as well as the model for Y as in Chapter 5. All simulations are run 100 times and results are averaged over these runs.

We run a simulation to illustrate the effect of the constraint, with results given in Fig. 6.1. Here we see an interesting result: while increasing the constraint parameter yields better results when we include the penalty term on the within-

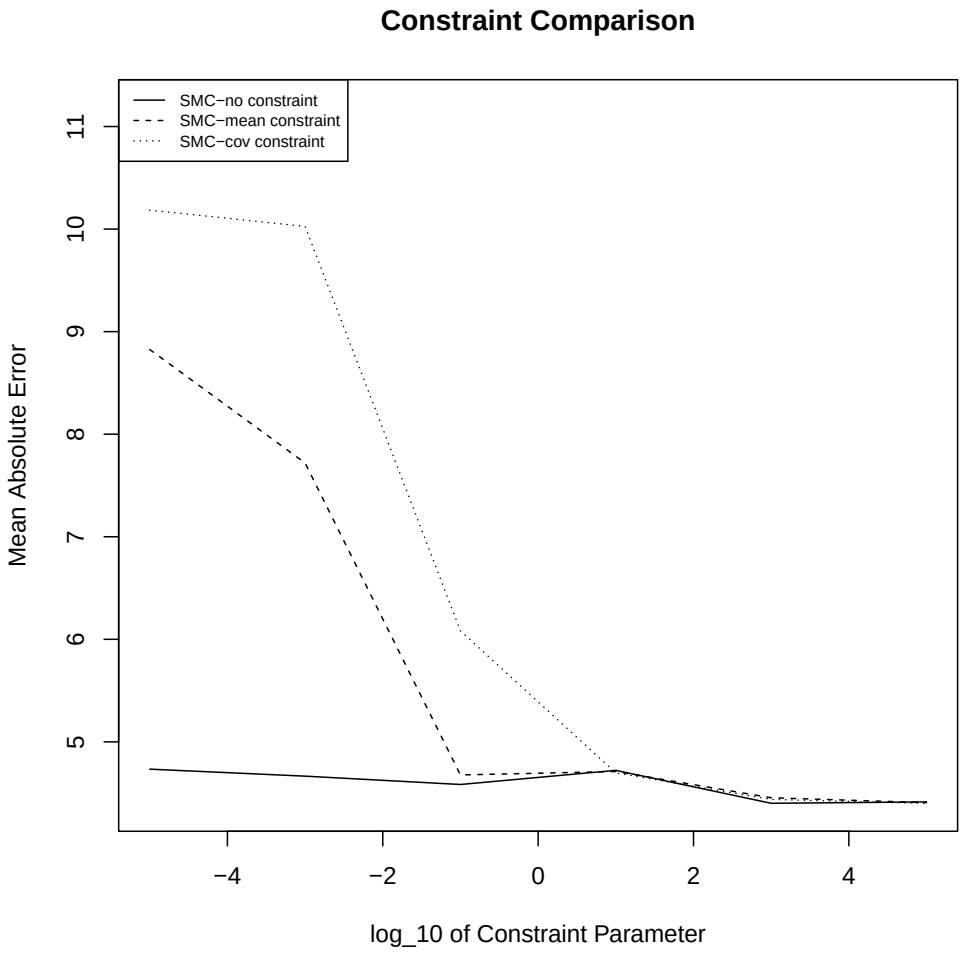


Fig. 6.1: Simulation results using various constraints.

class first or second moments, not including this penalty term at all seems to work best. This can be understood by the fact that we are effectively constraining our dimension reduction parameters in the initial terms

$$\frac{1}{\sigma_{ex}^2}||\,\mathbf{y} - \mathbb{X}\,\boldsymbol{\eta}\,\boldsymbol{\beta}\,||^2 + \frac{1}{\sigma_{ez}^2}||\,\boldsymbol{\xi} - \mathbb{Z}\,\boldsymbol{\gamma}\,\boldsymbol{\beta}\,||^2$$

through $\boldsymbol{\beta}$ in equation (6.5).

We run a simulation using the values $\alpha = 10^{-10}, 1, 10$ to illustrate how various

Table 6.2: Average mean absolute errors and standard errors from various simulation studies. "Truth" is taken to be a linear model estimated from the true values of $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$.

Method	$\alpha = 10^{-10}$	$\alpha = 1$	$\alpha = 10$
	Target MAE (SE)	**Target MAE (SE)**	**Target MAE (SE)**
"Truth"	22.44 (1.57)	2.22 (0.13)	0.48 (0.01)
PCR	22.52 (1.56)	2.62 (0.12)	12.13 (0.60)
KMM	25.40 (1.49)	2.82 (0.12)	11.90 (0.60)
KLIEP	22.55 (1.56)	2.61 (0.12)	12.12 (0.60)
IS	20.12 (1.27)	2.42 (0.10)	10.94 (0.48)
GFK	**19.53** (1.15)	2.47 (0.08)	11.35 (0.43)
RLD	22.56 (1.56)	2.61 (0.12)	9.57 (0.42)
SMC	20.20 (1.28)	**2.23** (0.11)	**5.46** (0.25)

alternative methods perform when models are highly nonlinear as well as highly linear. Results on the target data are given in Table 6.2. The estimates for "truth" use the values of $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$ that were used to generate the data. The quotes are to indicate that the model itself is still misspecified as a linear model, though the true values of the dimension reduction parameters are used. We see that SMC performs well in all cases, though it does not beat GFK or IS in the case of a more nonlinear model. Interestingly, when the model is close to linear, SMC vastly outperforms RLD, a similar method. This may be due to the fact previously stated, that SMC is effectively a version of RLD that uses the conditional model to relate the parameters $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$. Though it was not investigated, RLD may benefit from including a conditional term in its objective.

Table 6.3: Average mean absolute errors and standard errors for age estimation. The estimated dimension was taken to be 10. Minimum mean absolute errors are in bold.

Method	**Source MAE (SE)**	**Target MAE (SE)**
PCR	16.132 (0.098)	45.833 (8.686)
KMM	15.637 (0.212)	95.293 (26.548)
KLIEP	16.147 (0.098)	42.448 (8.546)
IS	7.448 (0.123)	28.008 (4.581)
GFK	**4.754** (0.078)	13.359 (0.383)
RLD	16.179 (0.099)	30.387 (4.999)
SMC	8.137 (0.093)	**11.919** (0.293)

6.5.2 Real Data

In addition to the simulation studies above we consider a real data example. We turn again to the FG-NET dataset, this time using landmark data points as source data and using the raw face data (rescaled to 10×10 grayscale images) as target. This is a potentially useful real-world application as it will often be difficult and time-consuming to obtain landmark points from an image on-the-fly given an unseen data point. In order for the dimension reduction methods to be used, we transformed the landmark data using the inverse exponential map as in Chapter 5 while leaving the raw face data untouched. Average mean absolute errors and their standard errors are given in Table 6.3. We see again that IS and GFK perform well, though only on the source data, with SMC not far behind. For the target data, SMC comes out ahead, with GFK performing competitively. As in the case of the simulation studies, RLD performs worse than SMC – significantly so – indicating again the possibility of improvement to RLD by incorporating the conditional error term.

6.6 Discussion

The SMC method can perform well on dimension reduction problems, though it seems that in many cases considered GFK performs competitively. While GFK has the added benefit of a small number of tuning parameters, the construction of the SMC estimates above have been somewhat primitive. We can extend the SMC method to incorporate multiple sources or multiple targets by considering a mixed model framework (see [75]) where the domain from which features come can be included as a random term in the model. Moreover, SMC can be extended to yield point estimates for $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$ through the posterior mode as opposed to the posterior mean. This can provide a framework from which to get a visual representation of the data, and can benefit practitioners desiring a more interpretable approach. Additionally, we can more easily influence the form of $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$; if sparse estimates are desired we can incorporate sparse priors, e.g., spike-and-slab priors [76]. This could lead to more useful and interpretable estimates as opposed to the potentially difficult-to-interpret kernel nearest-neighbor approach.

7

Discussion

7.1 Summary

Methods to handle prediction with high-dimensional inputs are invaluable to practitioners in modern data analysis. Many problems concerning high-dimensional data will be amenable to techniques to either incorporate a lower-dimensional structure implicitly – e.g., through penalized least squares – or explicitly - e.g., through linear dimension reduction. In regression problems, Tikhonov-style regularization can be used to construct estimators that take into account a lower-dimensional structure in the predictors, whether this structure is known a prior or not. For specific manifolds, such as the Grassmannian, a simple ridge regression can be used to significantly improve results.

Additionally, problems in which data arise from an inhomogeneous process crop up in many practical settings. The bulk of approaches concern either the estimation of weights to apply to the data that yield similarly distributed observations, or methods of dimension reduction to map data into spaces where observations are distributed similarly. Combined direction estimation seeks a linear dimension reduction related to incremental subspace estimation that can easily incorporate information from the conditional distribution of the response given the predictors. Moreover, "local" covariance structure can be estimated through assuming a non-

parametric error structure. This method can also easily be extended to the case of data lying on the Grassmannian.

In CDE, the use of the conditional information was through the specification of a model for the response given the predictors, and the dimension reduction parameters were obtained through a type of penalized least squares. In regularized likelihood directions, we attempt to incorporate this information directly into the predictors by assuming they are distributed as a mixture of normal distributions. In this case, we can specify a likelihood while incorporating a constraint for the within-class first moments to help take inhomogeneous data into account, in an attempt to incorporate a prior assumption on the conditional distribution of the response given the predictors.

Finally, Monte Carlo methods can be used to gain improvements in various settings, with sequential Monte Carlo methods being similar in spririt to many of the previous incremental subspace approaches. The Monte Carlo acquired directions framework uses all of the information about the conditional and marginal distributions while evolving the parameter estimates to ones that have useful properties.

7.2 Future Work

Many approaches we propose can be extended to yield potential improvements. While it was not considered in detail, sparse estimates can be obtained in CDE through alternating minimization, which could potentially yield more interpretable estimates. These sparse estimates could also provide improvements in prediction

when the true model is sparse. In the sequential Monte Carlo framework, we can extend the approach to problems with multiple data sources through considering a generalized linear model in which data sources are incorporated through a random error term [75]. Moreover, allowing the choice of priors (e.g., ones that are sparse) could be used to help achieve better or more interpretable results.

The current implementation of many of the approaches can be somewhat computationally intensive, with almost all approaches requiring a gradient descent-style method to obtain parameter estimates. More efficient implementations of these approaches could have the potential to greatly increase their utility. The methods could be adapted to "on-line" methods by introducing fidelity measures for new data points, classifying them as being from source, target, or uninformative [77].

Finally, nonlinearity has entered each of the proposed methods through local estimation. While this local approach improves performance, it greatly increases computation and moreover has the potential to be less interpretable. Investigating kernel approaches [78] or semiparametric techniques [68] to overcome some of this nonlinearity could prove worthwhile.

Problems with high-dimensional predictors abound in modern data analysis. Cases in which predictors correspond to visual information can yield improved performance through many techniques, such as through regularization while estimating a regression function to take into account underlying predictor structure, or through regularization of an objective function involving a dimension reduction parameter to improve prediction in the case of inhomogeneous distributions of data between training and testing. We see that incorporating penalties into various objectives can

yield estimates with desired properties in regression, classification, and dimension reduction, and consider these approaches vital to high-dimensional inference.

Bibliography

[1] John Burnet. *Early Greek Philosophy, 3rd edition*. A & C Black Ltd., London, 1920.

[2] Yasuko Chikuse. *Statistics on Special Manifolds*. Lecture Notes in Statistics. Springer, 2003.

[3] FG-NET Aging Database, Accessed Apr 2011. Face and Gesture Recognition Research Network, Available `http://www.fgnet.rsunit.com/`.

[4] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *European Conference on Computer Vision 2010*, Lecture Notes in Computer Science, pages 213–226. Springer, Berlin-Heidelberg, 2010.

[5] Richard Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[6] Charles J. Stone. Optimal Global Rates of Convergence for Nonparametric Regression. *The Annals of Statistics*, 10(4):1040–1053, 1982.

[7] Edward J. Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 85(411):664–675, 1990.

[8] John A. Thorpe. *Elementary Topics in Differential Geometry*. New York: Springer-Verlag, 1979.

[9] Anil Aswani, Peter J. Bickel, and Claire Tomlin. Regression on Manifolds: Estimation of the Exterior Derivative. *The Annals of Statistics*, 39(1):48–81, 2011.

[10] Kyle Gallivan, Anuj Srivastava, Xiuwen Liu, and Paul Van Dooren. Efficient algorithms for inferences on Grassmann manifolds. In *In Proceedings of 12th IEEE Workshop on Statistical Signal Processing*, pages 315–318, 2003.

[11] Ian T. Jolliffe. *Principal Component Analysis*. Wiley Online Library, 2005.

[12] R. Dennis Cook and Kofi P. Adragni. Sufficient dimension reduction and prediction in regression. *Philosophical Transactions of the Royal Society A: Physical, Mathematical and Engineering Sciences*, 367(1906):4385–4405, 2009.

[13] J. N. R. Jeffers. Two Case Studies in the Application of Principal Component Analysis. *Applied Statistics*, 16(3):225–236, 1967.

[14] Sam T. Roweis and Lawrence K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290:2323–2326, 2000.

[15] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science (New York, N.Y.)*, 290(5500):2319–23, December 2000.

[16] Baback Moghaddam. Principal Manifolds and Probabilistic Subspaces for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):780–788, June 2002.

[17] Gabriel Peyre. Manifold Models for Signals and Images. *Computer Vision and Image Understanding*, 113(2):249–260, February 2009.

[18] Daniel L. Rudermanli and William Bialek. Statistics of natural images: Scaling in the woods. *Physical Review Letters*, 1994.

[19] Jinggang Huang and David B. Mumford. Statistics of natural images and models. In IEEE Computer Society, editor, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 541–547. IEEE Computer Society Press, Los Alamitos, CA, 1999.

[20] Colin R. Goodall and Kanti V. Mardia. Projective Shape Analysis. *Journal of Computational and Graphical Statistics*, 8(2), 1999.

[21] Pavan Turaga, Soma Biswas, and Rama Chellappa. The Role of Geometry for Age Estimation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 946–949. University of Michigan Library, March 2010.

[22] Yun Fu and T. S. Huang. Human Age Estimation With Regression on Discriminative Aging Manifold. *IEEE Transactions on Multimedia*, 10(4):578–584, June 2008.

[23] Gaurav Aggarwal, Amit K. Roy Chowdhury, and Rama Chellappa. A System Identification Approach for Video-Based Face Recognition. *International Conference on Pattern Recognition*, 4:175–178, 2004.

[24] Hal Daumé III and Daniel Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26(1):101–126, 2006.

[25] Peter J. Bickel and Bo Li. Local polynomial regression on unknown manifolds. In Liu Regina, William Strawderman, and Cun-Hui Zhang, editors, *Complex Datasets and Inverse Problems: Tomography, Networks and Beyond*, volume 54, pages 177–186. IMS Lecture Notes, Monograph Series, 2007.

[26] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for non-orthogonal problems. *Technometrics*, 12:55–67, 1970.

[27] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[28] William F. Massy. Principal Components Regression in Exploratory Statistical Research. *Journal of the American Statistical Association*, 60(309):234–256, 1965.

[29] E. A. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964.

[30] T. Gasser, H-G. Müller, and V. Mammitzsch. Kernels for nonparametric curve estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, 47(2):238–252, 1985.

[31] Jianqing Fan and Irene Gijbels. *Local Polynomial Modelling and Its Applications*. Monographs on Statistics and Applied Probability. Chapman & Hall, 1996.

[32] Peter J. Bickel and Elizaveta Levina. Covariance regularization by thresholding. *Annals of Statistics*, 36(6):2577–2604, 2008.

[33] Baback Moghaddam and Alex Pentland. Probabilistic visual learning for object detection. In Massachusetts Institute of Technology, editor, *Proceedings of the Fifth International Conference on Computer Vision*, pages 786–793. IEEE Computer Society Press, Jun 1995.

[34] David J. Finney. *Probit Analysis*. Cambridge University Press, Cambridge, 1952.

[35] John A. Nelder and Robert W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society, Series A*, 135:370–384, 1972.

[36] Peter McCullagh and John A. Nelder. *Generalized Linear Models*. Chapman & Hall, London, 1989.

[37] Paul W. Holland and Roy E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics - Theory and Methods*, 6(9):813–827, 1977.

[38] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[39] Anil Aswani, Peter J. Bickel, and Claire Tomlin. Statistics for Sparse, High-Dimensional, and Nonparametric System Identification. *2009 IEEE International Conference on Robotics and Automation*, pages 2133–2138, May 2009.

[40] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. *IEEE Conference on Computer Vision and Pattern Recognition*, 1, 2001.

[41] Anil Aswani, Peter Bickel, and Claire Tomlin. Regression on Manifolds: Estimation of the Exterior Derivative. *Annals of Statistics*, to appear 2010.

[42] Xiaofei He and Partha Niyogi. Locality Preserving Projections. In Sebastian Thrun, Lawrence Saul, and Bernhard Scholkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, 2004.

[43] John A. Lee and Michel Verleysen. *Nonlinear Dimensionality Reduction*. Springer, 2007.

[44] Elizaveta Levina and Peter J. Bickel. Maximum likelihood estimation of intrinsic dimension. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA, 2005.

[45] Hidetoshi Shimodaira. Improving Predictive Inference Under Covariate Shift by Weighting the Log-Likelihood Function. *Journal of Statistical Planning and Inference*, 90(2):227–244, October 2000.

[46] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[47] Carl-Erik Särndal, Bengt Swensson, and Jan Wretman. *Model Assisted Survey Sampling*. Springer-Verlag, New York, 1992.

[48] James J. Heckman. Sample Selection Bias as a Specification Error. *Econometrica: Journal of the Econometric Society*, 47(1):153–161, 1979.

[49] Masashi Sugiyama, Taji Suzuki, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008.

[50] Jiayuan Huang, Alexander J. Smola, Arthur Gretton, Karsten M. Borgwardt, and Bernhard Schölkopf. Correcting sample selection bias by unlabeled data. *Advances in Neural Information Processing Systems*, 19:601, 2007.

[51] Yurii Nesterov and Arkadii Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM Studies in Applied and Numerical Mathematics, Philadelphia, PA, 1994.

[52] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain Adaptation for Object Recognition : An Unsupervised Approach. *International Conference on Computer Vision*, 2011.

[53] Herman Wold. Partial Least Squares. In S. Kotz and N. Johnson, editors, *Encyclopedia of Statistical Sciences*, pages 581–591. Wiley, New York, 1985.

[54] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[55] David W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, New York, Chichester, 1992.

[56] Ryan Rifkin and Aldebaro Klautau. In Defense of One-vs-All Classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

[57] Alan Edelman, T. A. Arias, and Steven T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Anaysis and Applications*, 20:303–353, 1998.

[58] Rabi Bhattacharya and Vic Patragenaru. Large Sample Theory of Intrinsic and Extrinsic Sample Means on Manifolds I. *Annals of Statistics*, 31(1):1–29, 2003.

[59] Pavan K. Turaga, Ashok Veeraraghavan, Anuj Srivastava, and Rama Chellappa. Statistical Computations on Grassmann and Stiefel Manifolds for Image and Video-Based Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2273–2286, 2011.

[60] Gene H. Golub, Michael Heath, and Grace Wahba. Generalized Cross-Validation as a Method for Choosing a Good Ridge Parameter. *Technometrics*, 21(2):215–223, May 1997.

[61] Clive R. Loader. Bandwidth Selection: Classical or Plug-In? *The Annals of Statistics*, 27(2):415–438, 1999.

[62] Ian L. Dryden and Kanti V. Mardia. *Statistical Shape Analysis*. Wiley, 1998.

[63] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *Computer Vision and Pattern Recognition*, pages 886–893. IEEE Computer Society, Washington, DC, 2005.

[64] R. Dennis Cook and Liliana Forzani. Likelihood-based sufficient dimension reduction. *Journal of the American Statistical Association*, 104(485):197–208, 2009.

[65] David A. Shaw and Rama Chellappa. Combined Direction Estimation for Dimension Reduction in the Presence of Inhomogeneous Data. *Journal of the American Statistical Association*, 2014. Under review.

[66] C. Radhakrishna Rao. Separation theorems for singular values of matrices and their applications in multivariate analysis. *Journal of Multivariate Analysis*, 9(3):362–377, September 1979.

[67] David A. Shaw and Rama Chellappa. Regression on Manifolds Using Data-Dependent Regularization with Applications in Computer Vision. *Statistical Analysis and Data Mining, Special Issue: Joint Statistical Meetings 2012*, 6(6):519–528, December 2013.

[68] Yanyuan Ma and Liping Zhu. A Semiparametric Approach to Dimension Reduction. *Journal of the American Statistical Association*, 107(497):168–179, 2012.

[69] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 04 2004.

[70] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse Principal Component Analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, June 2006.

[71] Ker-Chau Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(141):316–327, 1991.

[72] Ross A. Lippert and Alan Edelman. SGMIN. `http://www.cs.ucdavis.edu/~bai/ET/other_methods/overview_SGMIN.html`, Accessed Dec 2012.

[73] David A. Shaw and Rama Chellappa. Sufficient dimension reduction for domain adaptation, 2014. Submitted to European Conference on Computer Vision.

[74] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10:197–208, 2000.

[75] Y. Fan, D. S. Leslie, and M. P. Wand. Generalised Linear Mixed Model Analysis via Sequential Monte Carlo Sampling. *Electronic Journal of Statistics*, 2:916–938, 2008.

[76] Hemant Ishwaran and J. Sunil Rao. Spike and slab variable selection: Frequentist and bayesian strategies. *The Annals of Statistics*, 33(2):730–773, Apr. 2005.

[77] Alex J. Smola, S. V. N. Vishwanathan, and Thomas Hofmann. Kernel methods for missing variables. In Robert G. Cowell and Zoubin Ghahramani, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, Jan 6-8, 2005, Savannah Hotel, Barbados*, pages 325–332. Society for Artificial Intelligence and Statistics, 2005.

[78] Samuel Gerber, Tolga Tasdizen, and Ross Whitaker. Dimensionality reduction and principal surfaces via kernel map manifolds. *International Conference on Computer Vision*, pages 529–536, 2009.