Semi-supervised Subspace Learning and Application to Human Functional Magnetic Brain Resonance Imaging Data

Eberhard Karls Universität Tübingen
Fakultät für Informations- und Kognitionswissenschaften
Wilhelm-Schickard Institut für Informatik

Master Thesis in Computer Science

by

Jacquelyn A. Shelton

Prof. Dr. A. Schilling
Department of Informatics
Wilhelm-Schickard-Institute
at the Eberhard Karls University of Tübingen

Dr. Matthew B. Blaschko
Department of Empirical Inference
Max-Planck-Institute
for Biological Cybernetics Tübingen
and the University of Oxford

July 2010

Selbständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe und dass alle Stellen, die dem Wortlaut nach anderen Werken entnommen sind, durch Angaben von Quellen als Entlehnung kenntlich gemacht worden sind. Diese Masterarbeit wurde in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt.

Ort, Datum Unterschrift

The work described in this thesis took place at the Max Planck Institute for Biological Cybernetics, in the department of Empirical Inference under Bernhard Schölkopf and the department of Neurophysiology under Nikos Logothetis. I would like to thank Matthew Blaschko and Andreas Bartels for supporting through this project, for providing functional magnetic resonance imaging data and fruitful discussions. A special thanks to Christoph Lampert for taking the time to ensure my understanding of a topic or derivation, and to both he and Arthur Gretton for always humoring my many questions. Finally, thanks to Andreas Schilling for supervision of this project.

Contents

Chapter 1

Introduction

Machine learning is a modern and rapidly growing empirical science which integrates themes in statistical inference and decision making with a focus on exploratory data analysis using computational methodology (Bishop (2006)). It is a branch of artificial intelligence which draws strongly upon methodology from linear algebra, optimization, and signal processing in order to develop and apply predictive models originating from statistics, computer science, and engineering. The main concern of machine learning is to design algorithms which enable machines to *learn*, particularly by finding patterns or regularities in data and using these patterns to drive some decision/analysis process (Bishop (2006); Duda et al. (2001); Schölkopf and Smola (2002)).

1.1 Motivation

There has been a recent interest in neuroscience in assessing natural visual processing, by measuring the activity in the brain that occurs during a natural visual processing task. This is often approached via experiments in human functional magnetic resonance imaging (fMRI), wherein the human volunteers are instructed to lie in an fMRI scanner as they are shown some natural video stimulus (such as a movie clip) and the corresponding brain activity is simultaneously recorded. The goal of these studies is the infer the active regions of the brain during some aspect of the natural visual stimuli, such as during the presentation of human faces.

These fMRI studies are plagued with a number of problems which are very well-adapted for advanced machine learning methods. First, the *data are very high-dimensional* brain images. As such, in order to infer which areas of the brain are active, one needs to employ a dimensionality reduction method to reduce the overall activity to localized regions in the brain. Second, fMRI analysis is inherently a data poor domain. There are relatively very few corresponding time samples of the fMRI brain images with respect to the dimensionality of the data (e.g. limited time a human can remain safely in an fMRI scanner and high demand on fMRI facilities). This renders the problem *susceptible to overfitting*, which will be discussed below. Finally, given that the goal is to assess *natural* visual processing via natural visual stimuli, one *needs expensive manually-made labels* indicating the content of the stimuli. This is a very time-consuming and expensive process, requiring approximately five human observers to label every few frames of a stimulus with the degree to which a set of 'features' are present (i.e. human faces, human bodies, etc.). These labels are used to infer and localize the active brain regions occurring during the presentation of these features and are crucially important in fMRI studies that seek to localize stimulus-driven brain activity.

This thesis proposes a novel method in machine learning motivated by a new application to a problem in neuroscience. To begin, we will discuss some basic machine learning concepts.

1.2 Learning

Broadly speaking, *learning* is generally referred to as a method that incorporates information from a sample of n-observations $X = \{x_1, \ldots, x_n\}$, called the *training data* to construct an algorithmic model of the generalities in the data (i.e., regularities and/or structure), and then to infer some decision or prediction (i.e. classification). In statistics, a similar process is referred to as estimation. The main goal in machine learning is to construct a learning model with as few assumptions on the data as possible, on the training data, which thus allows the model to make reasonable decisions/predictions about unseen data, on *testing data*. The form of such an algorithm can be expressed as a function $f(x)$, where f takes the training data X as input, and produces an output f encoded in the same way as X.

The nature of $f(x)$ is generally determined by the *learning phase* or *training phase*, where the regularities are quantified and the form of the decision/prediction is made based on the training set X. Now the algorithm can be tested on new, unseen data, the testing data, where we can observe how it performs in practice. The ability of $f(x)$ to generalize its inferential predictions to unseen data is called *generalization* – the trade-off between the performance of $f(x)$ on the training set versus its performance on testing sets (generalizability) is an important focus of research in machine learning (Vapnik, 1998).

The learning algorithm can be constructed in various frameworks, in which various degrees of feedback are provided. First, the method can be in a *supervised learning framework*, in which there is a set of target *labels* $Y = \{y_1, \ldots, y_n\}$ corresponding to each $\{x_1, \ldots, x_n\}$ which indicates the target value of the ith sample of x. When Y consists of discrete values, the problem is called *classification*, whereas when Y consists of continuous values, the problem is called *regression*. The process of obtaining a set of labels Y is generally laborious and expensive. Second, the learning can be in a *unsupervised framework*, where there is no set of labels to guide/supervise the learning phase, instead the learning goal is to let the data itself define the structure (similarities or distribution) discovered. Finally, the *semi-supervised learning framework* is a combination of the two above approaches, allowing inclusion of data samples with and without labels (reducing the need for as many of these expensive labels) for an increased data set that generally leads to an improved predictive performance.

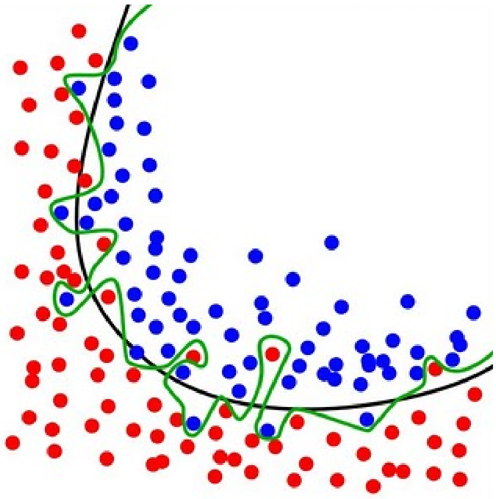
1.2.1 Overfitting and Regularization

As mentioned, the performance of a learning model on testing data, how well it generalizes, needs to be balanced with the model performance on the training data. Performance is evaluated by some quantified term, e.g. by a loss function quantifying the error rate, and as such the goal is to learn e.g. a function $f(x)$ with minimal testing error balanced with minimal training error (thereby minimizing the loss function). When the model is too catered to the testing data, the function $f(x)$ is likely to be too complex to be able to also yield good performance on the testing data. This situation known as *overfitting*.

Main contributions to overfitting are high-dimensionality of data with a relatively low number of data samples. The reason behind this is that there are too many degrees of freedom for the learning function $f(x)$ to adapt to the data samples. Thus, when there are relatively few data samples available with respect to the dimensionality of the data, $f(x)$ will be highly catered to the data samples, and correspondingly too complex to yield good generalization performance.

To reduce overfitting, one has the option of *regularization* (Tikhonov and Arsenin (1977); Tikhonov (1963)), which seeks to balance the trade-off between the testing error and training error. Regularization is particularly important in the case of high-dimensional data. Regularization introduces parameters to the learning model that penalize complexity of the learned function, which leads to the favoring of smooth functions in order to increase generalizability of $f(x)$. The basic idea behind this is that, when a learned model is perfectly catered to the given training data set, e.g. with training error equal to 0, then this will be some complex function with a solution that is very unlikely to be generalizable to a new data set, and thus will yield high testing error. See Figure 1.1 for an illustration of

$f(x)$ in the case of learning a simple 2-class classification function, and Figure 1.2 for a possible corresponding plot of the types of learning error in this scenario.



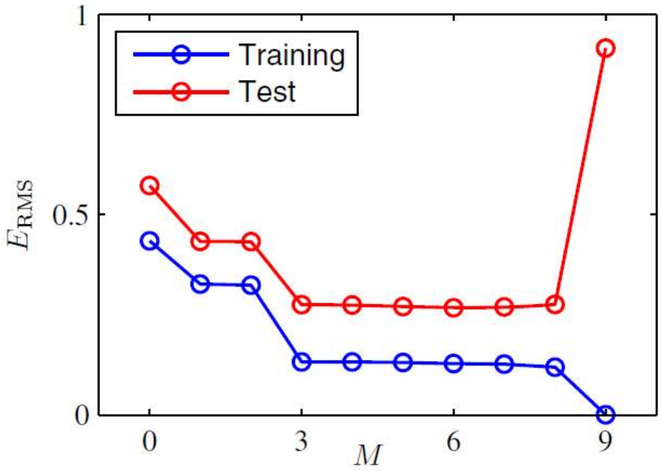
(a) Complex 2-class classification function versus a regularized, smooth classification function.

Figure 1.1: *This figure illustrates two classes of objects, the red and the blue, with two different discriminant functions learned from the same model but with different degrees of regularization. The green curve illustrates a function $f(x)$ that is unregularized – it perfectly discriminates the two classes, but the function is quite complex. The black curve represents a regularized function which favors a smooth discriminant function – it accepts some amount of training error for the sake of being more generalizable to unseen data (yielding a lower testing error). (Figures adapted from Bishop (2006))*

The regularization parameters of the learning model are selected through a process called *model selection* (e.g. Bishop (2006)), which seeks the best parameters with minimal overfitting and maximal generalizability. One way to do this is via cross validation, which partitions the data set into training and testing sets, and trains the model on different variations of these divisions. Other methods include data sampling and computational heuristics (e.g. Hardoon et al. (2004)).

1.3 Outline

Canonical Correlation Analysis (CCA; see Chapter 2) is a supervised dimensionality reduction technique that can take advantage of data available in multiple modalities (multiple forms). In the setting described above for instance, when data is available in the form of fMRI recordings and the expensive man-made labels, CCA is a good method of choice



(a) Training error versus testing error.

Figure 1.2: *As an intuitive example, this figure illustrates the general trend of testing error versus training error as a function of parameter M, which characterizes the complexity of the learned discriminant function $f(x)$. As the complexity of the learned function increases, the testing error increases (overfitting) and the training error drastically decreases, whereas when the function is relatively smooth (with some amount of regularization enforcing smoothness), then the trade-off between the error types is balanced. In other words, overfitting is balanced with generalizability. (Figures adapted from Bishop (2006)).*

because it can utilize both modalities to drive the search for a lower-dimensional representation of the fMRI data, as opposed to well-known methods such as Principle Components Analysis (PCA) which can use only single-modality data. Furthermore, CCA has been recently introduced (Blaschko et al. (2008)) in a *semi-supervised learning framework* which uses an additional form of regularization in order to include the unlabeled data in the learning problem. Prior to this thesis, this method has not been applied on real data, and by utilizing CCA in a semi-supervised learning framework, we can make maximum usage of *all available fMRI data*. In other words, even if we do not have labels corresponding to all stimuli shown to the human volunteers, but rather we only have the fMRI activity recorded during the unlabeled stimulus' presentation, we can still include this data in our semi-supervised dimensionality reduction framework. This method and framework will thus allow us to reduce the necessity of these costly stimulus labels, and at the same time increase our data samples under consideration and reduce overfitting.

We propose to approach the problem of stimulus-driven dimensionality reduction of fMRI data with the novel method of semi-supervised kernelized CCA using Laplacian regularization.

This thesis will begin with an introduction to supervised and unsupervised dimensionality reduction methods, including the special cases of CCA, in Chapter 2. These linear methods will then be generalized to account for nonlinear patterns in data in Chapter 3. Next, in Chapter 4 we will discuss the semi-supervised learning framework and how to incorporate the method of CCA in this framework. In Chapter 5 we will become familiar with the methods and types of fMRI data used in the experimental sets and formalize the entire experimental setting. The two sets of novel learning experiments and their evaluation methods will be described in detail with their corresponding results in Chapter 6. Finally, in Chapter 7 we conclude the thesis with a discussion of these results.

This thesis assumes familiarity with linear algebra and basic probability theory. These concepts and the corresponding notation used through the thesis can be found in the Appendix.

Chapter 2

Dimensionality Reduction

Most real world data, such as speech signals, digital photographs, and functional magnetic resonance imaging data scans, are extremely *high-dimensional*: they consists of measurements of the natural world which can be be represented as vectors consisting of thousands of dimensions. High-dimensional data carry however many burdensome properties. First, the data is particularly expensive in terms of storage; the size of which increases linearly with each dimension. Second, it cannot be easily visualized; it is difficult to understand what information high-dimensional contains. Also, due to the curse of dimensionality (Bellman (1961)) – which states that the volume of the subspace increases exponentially with the number of dimensions – the search of high-dimensional spaces is very computationally expensive. As such, manipulations and analyses of data lying in such a space is cumbersome in terms of storage and computational costs. Thus for various reasons of feasibility in dealing with data of such high-dimensions, reducing its dimensionality is desirable.

Dimensionality reduction techniques can be employed to reach many goals in machine learning. Depending on the main application, the dimensionality-reduced representation can be utilized for compression/reconstruction, visualization, feature extraction and classification, as well as regression, the latter of which is most related to the problems to be discussed in this thesis. As the name indicates, dimensionality reduction techniques seek to find the lower dimensional representation that is somehow intrinsic to a given data set, i.e. capturing its ”interesting structure”. The purpose is to find a lower-dimensional representation which removes the ”less important” dimensions/bases, ones capturing information deemed unimportant by the optimization criterion of the given method (i.e. when the projection of 3-dimensional data onto 2-dimensions, captures some acceptable percentage of the variance in data). Corresponding to this representation, the transformation function (projection) to this reduced basis set (subspace) is sought. See the illustration in

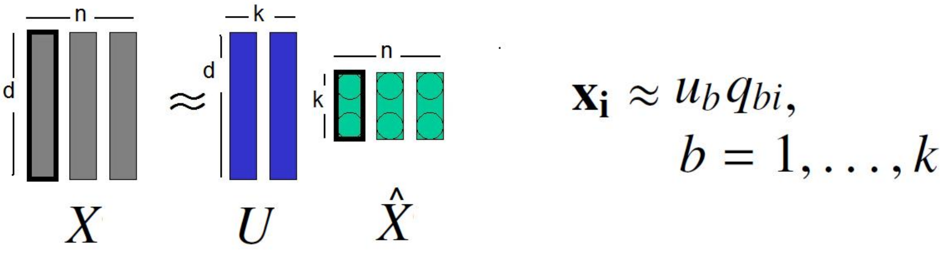


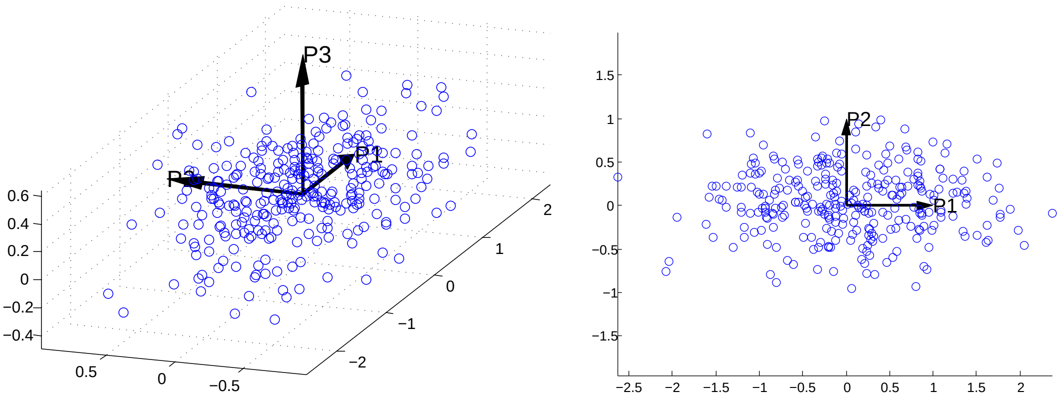
Figure 2.1: *Dimensionality reduction methods manipulate the selection of U, which can be used to project X to a lower-dimensional subspace. This entails selecting important basis functions from U via a reduced set of coefficients in Q.*

Figure 2.1.

Despite the typically high dimensionality of natural data, their intrinsic complexity and local dimensions are generally much lower, given that constraints within the natural world and through the imaging process of acquiring the images leads the data to occupy a lower-dimensional subspace. Specifically, as mentioned, the goal is to learn the 'best' geometric representation of a data set by finding a set of basis vectors of a more suitable subspace. The criterion for 'best' is characterized differently for each dimensionality reduction method, and defined by a corresponding objective function which, upon optimization will yield this criterion-defined lower-dimensional representation.

Now we will formalize the above setting of dimensionality reduction. Let $\mathcal{V}$ be an $\mathbb{R}$-vector space and $\mathcal{U} \subseteq \mathcal{V}$. $\mathcal{U}$ is called a subspace of $\mathcal{V}$ if and only if for all $u_1, u_2 \in \mathcal{U}$ and for all $\lambda_1, \lambda_2 \in \mathbb{R}$ the following holds: $\lambda_1 u_1 + \lambda_2 u_2 \in U$. The goal is to identify some $\mathcal{U}$ that is of *fewer* dimensions than $\mathcal{V}$ in which we can more efficiently represent and handle the data set. Formally this is written $X \approx UQ$, where $X = (x_1, \dots, x_n) \in \mathbb{R}^{d \times n}$, then we look for $u_1, \dots, u_k$ basis vectors ($U \in \mathbb{R}^{d \times k}$) of the subspace $\mathcal{U}$, and $\hat{X} \in \mathbb{R}^{k \times n}$ is the set of coefficients representing X. Notice that when $k = d$, this is only a change of base and the approximation $X \approx U\hat{X}$ in Figure 2.1 becomes the equivalence $X = U\hat{X}$. Furthermore, since U is the matrix of the basis vectors of $\mathcal{U}$, and $U^T U = I_{k \times k}$, it follows that $\hat{X} = U^T X$. Different dimensionality reduction methods have a different task-dependent criterion for the 'best' selection of U, but the end goal is to have a smaller set of 'important' coefficients $\hat{X}$ using correspondingly fewer basis vectors U, i.e. to represent the data in the *lower-dimensional coordinate system* of U (refer again to Figure 2.1).

In this chapter I will discuss a few different but related techniques often used for dimensionality reduction, all of which may be mathematically expressed as a generalization of Principle Components Analysis (PCA).



(a) Principle directions.

(b) Projected data.

Figure 2.2: *Example of dimensionality reduction using Principle Components Analysis (PCA). (a) Directions within the data set capturing the maximum variance; three principle components of the data found by PCA. (b) Projection of the data onto the first two principle components (reduction to two dimensions).*

2.1 Principle Components Analysis (PCA)

Principle Components Analysis (PCA) is a traditional approach to reducing the dimensionality of a data set (Hotelling (1936)). It is an unsupervised method that finds the lower-dimensional representation of the data (the basis vectors of the subspace) that preserves most the data's variance. For this, it uncovers the '*principle components*' of the data set, the orthogonal directions that are aligned with the directions of maximal variance of the data (Figure 2.2 (a)), and projects the data onto a select number of these components (Figure 2.2 (b)). PCA is very useful as a compression method for a compact representation of a data set, as the number of directions that align with high variance is generally much less than the original number of dimensions, and with which the data can be well reconstructed.

To formalize PCA, consider the data set $X = (x_1, \ldots, x_n) \in \mathbb{R}^{d \times n}$, for which we assume the mean ($\frac{1}{n}\sum_{i=1}^{n} x_i = 0$) has been subtracted from each sample x_i. As a dimensionality reduction method, PCA seeks a matrix U with which to project the data X such that $\hat{X} = UX$, where $\hat{X}$ is the X data projected via U. PCA requires that the projection direction U is such that the projected data has maximal variance. From now on we will denote all quantities with a hat that are the projections of the original data. In the case of seeking only the first principle component of X, we want to find the u_1 where the variance of $\hat{X}$ is maximized (in $\hat{X} = u_1 X$), such that the norm of u_1 is 1 ($\|u_1\| = 1$):

$$\begin{aligned}\max_{u_1} var(u\hat{X}) &= \max_{u_1} \sum_{i=1}^{n} \hat{x}_i \hat{x}_i && (2.1)\\ &= \max_{u_1} \sum_{i=1}^{n} (u_i X)(u_i X)^T && (2.2)\\ &= \max_{u_i} u^T X X^T u && (2.3)\end{aligned}$$

subject to the constraint:

$$\|u\| = 1 \tag{2.4}$$

Now instead of XX^T above, we use the normalized covariance matrix of X:

$$C_{XX} = \frac{1}{n} X X^T. \tag{2.5}$$

With which we can write the optimization problem by *maximizing* the following *objective function*:

$$\textit{PCA objective function} \qquad u^T C_{XX} u, \tag{2.6}$$

subject to:

$$\begin{aligned}\|u\| &= 1 && (2.7)\\ \Leftrightarrow \|u\|^2 &= 1 && (2.8)\\ \Leftrightarrow u^T u &= 1 && (2.9)\\ \Leftrightarrow u^T u - 1 &= 0 && (2.10)\end{aligned}$$

For optimization of Expression (2.6) we can use Lagrangian formalism (e.g. Burges (2004)), which leads to the following Lagrangian function:

$$L(x, \lambda) = u^T C_{XX} u - \lambda(u^T u - 1), \tag{2.11}$$

where $\lambda \in \mathbb{R}$ is a Lagrange multiplier used to enforce the constraint. The minimization of L is:

$$\begin{aligned} \frac{\partial L}{\partial u}(u,\lambda) &= 0 && (2.12)\\ \Leftrightarrow 2C_{XX}u - 2\lambda u &= 0 && (2.13)\\ \Leftrightarrow C_{XX}u &= \lambda u, && (2.14) \end{aligned}$$

where the last expression in (2.14) is the eigenvalue problem to be solved. The maximization of the PCA objective function in Expression (2.6) has the corresponding value:

$$u^T(C_{XX}u) = u^T(\lambda u) \qquad = \lambda u^T u = \lambda. \tag{2.15}$$

Consequently we find the first principle component u_1 by computing the eigenvalues of C_{XX} and selecting the eigenvector u_i with the largest corresponding eigenvalue λ_i. Thus as u_1 is the first direction corresponding to the maximal variance of the projected data, we can use similar calculations to acquire the subsequent directions of maximal variance. We require each subsequent direction to be orthogonal to those previous, which is identical to sorting the remaining eigenvectors by their eigenvalues (e.g. Duda et al. (2001)).

The number of principle components selected from U for projection of X is the new reduced dimensionality of $\hat{X}$; in other words, the original n-dimensional data X is projected onto $d(\ll n)$-dimensional linear subspace spanned by these top eigenvectors/principle components. If the data to be considered by PCA did indeed lie in a linear subspace, the method is guaranteed to discover the subspace's dimensionality and thereby produce a compact representation.

2.1.1 Properties of PCA

The objective function of PCA in Expression (2.6) can be equivalently written as the following *Rayleigh quotient* optimization problem:

$$\textit{PCA objective function} \qquad \frac{u^T C_{XX} u}{u^T u}. \tag{2.16}$$

which is a form of optimization problem that can be solved as an eigenvalue problem, and generalized eigenvalue problems as we will see with the next dimensionality reduction techniques.

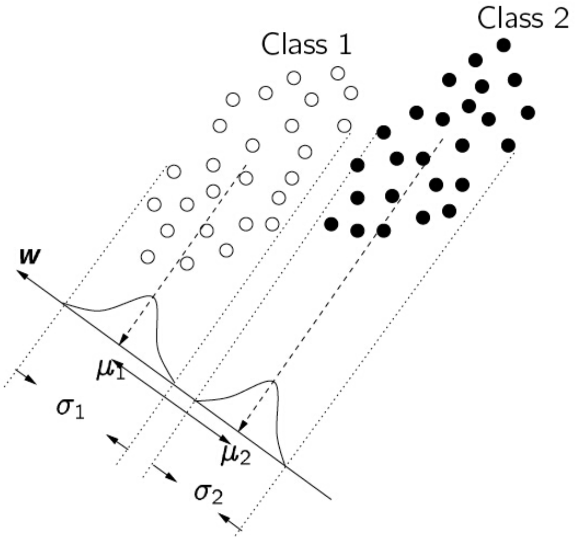


Figure 2.3: *LDA finds the most discriminative vector directions, projection of the data upon which, most separates the data classes. (Figure courtesy of Elisabeth Georgii, PhD Tutorial, 2007.)*

The resulting representation of the data after projection onto the principle components, $\hat{X}$, is decorrelated and has a diagonalized covariance matrix with the eigenvalues along the diagonal. Redundancies in the data have been removed by projection onto the eigenvectors of the covariance matrix, U.

Limitations include that PCA only considers the variance in the data and which corresponds to an implicit assumption that the data is generated from a Gaussianity distribution, which in practice is very unlikely. Furthermore this corresponds to the (often faulty) assumption that the underlying structure of the data can be represented via the variance. This property of PCA also renders it sensitive to outliers, as outliers in a signal will manifest in high variance. Additionally, PCA can only consider one domain of data at a time. As it is unsupervised, it can make no use of labels even if available, a potential drawback which as we will see, the other methods to be discussed can avoid. Finally, PCA can only discover linear patterns in the data, a limitation faced by all the dimensionality reduction methods to be discussed in this chapter. Non-linear alternatives will be discussed in Chapter 3.

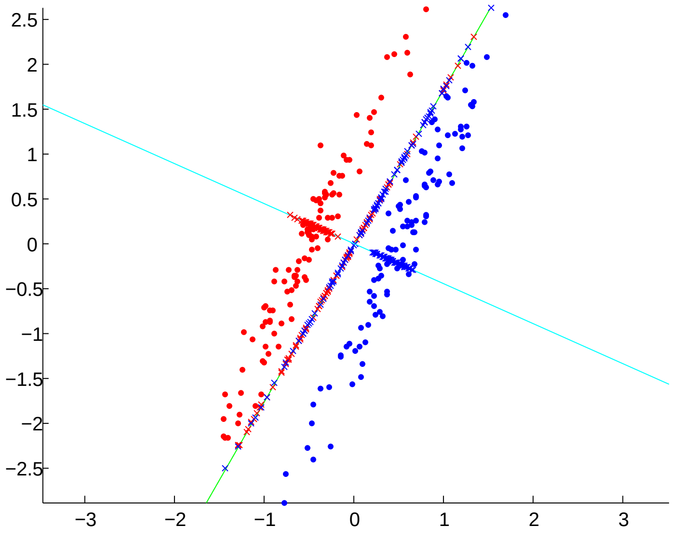


Figure 2.4: *PCA versus LDA. The green line is the projection direction found by PCA (first principle component), whereas the cyan line is the projection direction found by LDA.*

2.2 Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is a supervised dimensionality reduction technique, generally used either in classification or as a pre-processing feature extraction step (Belhumeur et al. (1997)). For classification, LDA is employed to find the best representation to separate classes within the data, resulting in its projection onto a 1-dimensional subspace. When used for feature extraction, the resulting separated-class projection can be used for i.e. later classification. As such, the goal of LDA is to discover a direction in the data upon which to project the data to maximally discriminate its classes. This low-dimensional projection of the data is accomplished by: 1) maximizing the variance between the classes (points belonging to different classes are far from each other) and 2) minimizing the variance within the classes (points belonging to same classes are near to each other). See Figure 2.3 for an illustration of such a projection. In this example, if the data were projected on the original x- or y-axis, the classes would intermingle, however they are not intermingling on the projection line, but instead are decently distinguished.

For illustration of LDA, we will consider the 2-class case. Consider once more the sample of $X = (x_1, \ldots, x_n) \in \mathbb{R}^{d \times n}$ to which we have the class label information of $Y = (y_1, \ldots, y_n)$ with $y_i \in \{-1, +1\}$. We denote the data belonging to the positive and negative classes as $X^+ = \{x_i : y_i = +1\}$ and $X^- = \{x_i : y_i = -1\}$, and data after projection as $\hat{X}^+$ and $\hat{X}^-$. Instead of the sample variances themselves, we define the squared distance between the 1-dimensional projected class sample means as the *scatter* between the classes as follows:

$$S_B = (m^+ - m^-)(m^+ - m^-)^T, \tag{2.17}$$

where S_B is the *between-class scatter matrix* between the classes, and m^+ and m^- are the sample means for the projected points (the projection of the sample points corresponding to that class label). Furthermore we define the scatter within the classes in the sample as the squared distance between the class means:

$$\begin{aligned} S_W &= \sum_{X^+=\{x_i:y_i=+1\}} (x_i - m^+)(x_i - m^+)^T + \sum_{X^-=\{x_i:y_i=-1\}} (x_i - m^-)(x_i - m^-)^T \qquad (2.18) \\ &= S^+ + S^- \qquad (2.19) \end{aligned}$$

where S_W is the *within-class scatter matrix* and S^+ and S^- are the scatter matrices for the classes individually.

As the optimization goal is to find a projection u which maximizes the separation of the classes, we want to maximize the ratio of these two numerical quantities (ratio of between-class and within-class scatter), which will yield the maximal separation of the classes. In order to work towards the formulation of this optimization goal, we rewrite these scatter formulations (Equations (2.17) and (2.18)) as their projections onto the direction u beginning with the between-class scatter:

$$\begin{aligned} \hat{S}_B &= (u^T m^+ - u^T m^-)^2 \qquad (2.20) \\ &= u^T (m^+ - m^-)(m^+ - m^-) u \qquad (2.21) \\ &= u^T S_B u \qquad (2.22) \end{aligned}$$

and the within-class scatter:

$$\hat{S}_W = u^T S_W u. \tag{2.23}$$

When label information corresponding to the class data is present, LDA can often outperform PCA (see Figure 2.4) (Blaschko and Lampert (2008)). However, solving LDA entails estimating the covariance matrix for each class, thus implying a need for larger sample size in order to avoid overfitting. When the sample size is insufficient with respect to the dimensionality of the data, LDA overfits to the data and PCA generally offers better performance (Andersen and Martinez (2001)). This is an important topic for discussion in later sections.

These matrices S_B and S_W are related to the cross-covariance matrix of the samples, and the covariance matrix for combined class samples (S^+ and S^-), respectively. To note is that both matrices are symmetric and positive semidefinite, important properties when solving generalized eigenvalue problems.

Thus the optimization problem for LDA can be expressed by *maximizing* the following *objective function* with respect to u:

$$\textit{LDA objective function} \qquad \frac{u^T S_B u}{u^T S_W u}, \tag{2.24}$$

which we can equivalently rewrite as the maximization of this constrained optimization problem:

$$u^T S_B u \tag{2.25}$$

subject to:

$$\tag{2.26}$$

$$u^T S_W u = 1, \tag{2.27}$$

and can solve as a generalized eigenvalue problem, by employing the same method of Lagrangian formalism as with the objective function of PCA in Equation (2.6). The Lagrangian function for LDA is:

$$L(u, \lambda) = u^T S_B u - \lambda(u^T S_W u - 1), \quad \lambda \in \mathbb{R} \tag{2.28}$$

where λ is a Lagrange multiplier used to enforce the constraint. The minimization of L is:

$$\frac{\partial L}{\partial u}(u, \lambda) = 0 \tag{2.29}$$

$$\Leftrightarrow 2S_B u - 2\lambda S_W u = 0 \tag{2.30}$$

$$\Leftrightarrow S_B u = \lambda S_W u, \tag{2.31}$$

where the last Expression (2.31) is a generalized eigenvalue problem.

The solution in Equation (2.31) yields the direction u upon which the projection of the data will be maximally separated, the classes maximally discriminated. Thus the original d-dimensional data set X is now reduced to, in this case, the 1-dimensional discriminative subspace. As u is the first direction for the projection $\hat{X}$ that corresponds to maximal separations of the classes in X, we can use similar calculations as in PCA to acquire the next directions, by searching for orthogonal directions, the next generalized eigenvectors.

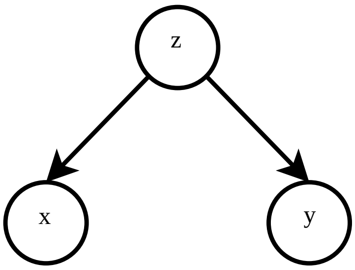
2.2.1 Properties of LDA

The afore described 2-class case of LDA can be generalized to data containing k-classes. Instead of estimating the scatter matrices S_B and S_W for only 2 classes, they are estimated for the k different classes exactly as they were in Equations (2.17) and (2.18), respectively.

2.3 Canonical Correlation Analysis (CCA)

Canonical Correlation Analysis (CCA) is a very general subspace method. It is a supervised technique that is closely related to both of the previous techniques of PCA and LDA. It is a dimensionality reduction technique that can take advantage of data in two domains, in contrast to both PCA and LDA, which can consider only one. As such, it searches simultaneously for projection directions in both data spaces in question such that the correlation is maximized between said projection vectors in each space. CCA is used for a broad range of applications: from information retrieval (Hardoon et al. (2004)), to multi-modal data clustering (Blaschko and Lampert (2008)), to independence tests (Hardoon et al. (2007)).

CCA takes advantage of data samples that are available in multiple modalities, specifically, we assume the data composes various views of some latent process. See Figure 2.5 for an illustration of this type of process and the resulting data samples. Because the available data samples (i.e. X and Y) are assumed to be representations of the same process (i.e. Z), this suggests a relationship between the data sets. Because CCA seeks to maximize correlation between projections in both X and Y, this induced dependence from Z allows CCA to utilize information in both variables to drive the discovery of the underlying structure in Z. This property also implies the close relation between CCA and mutual information (Borga and Borga (1998)). CCA finally yields a dimensionality-reduced representation (projection) of Z in each modality X and Y that are maximally correlated with each other.



(a) Paired data.

Figure 2.5: *With paired data, there are two observed output variables, x and y, which are generated by some underlying process z. Given that x and y are assumed to be stemming from the same underlying process, this induces a dependence between x and y.*

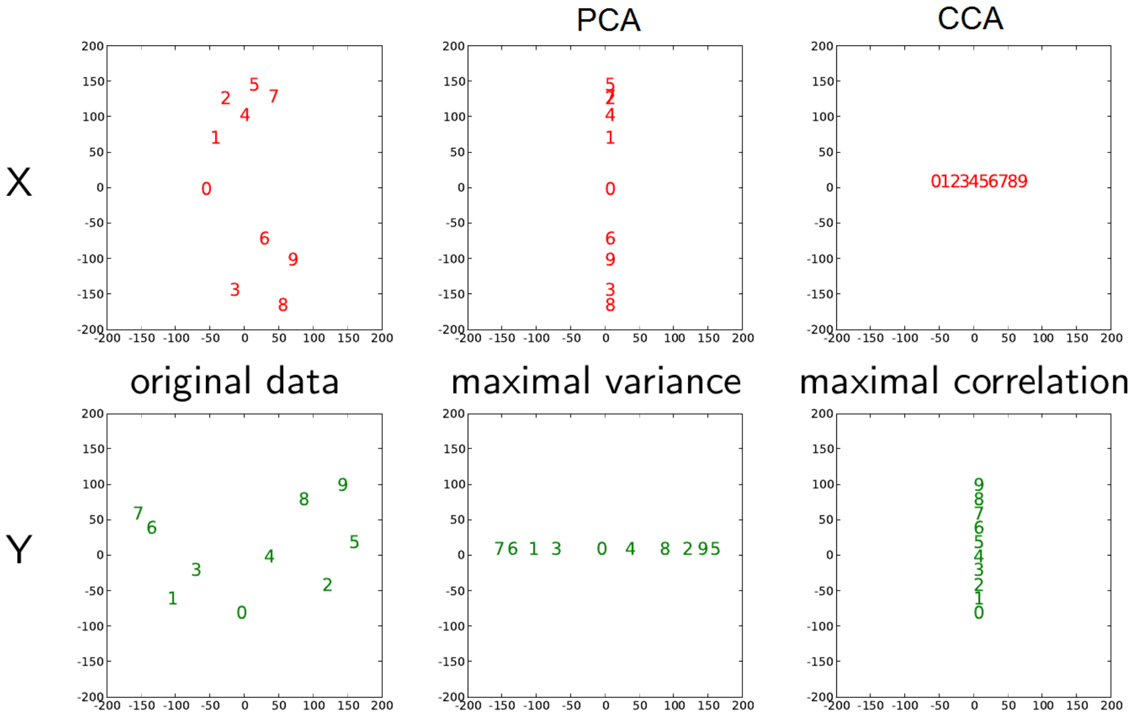


Figure 2.6: *PCA versus CCA. We have two noisy samples of a signal, the data modalities X and Y, with pairings between samples as indicated by the number of the sample point. The directions found from PCA (center) and CCA (right) demonstrate CCA's ability to ignore the non-correlated directions of high-noise in both modalities, yielding instead directions of the signal in each. (Figures adapted courtesy of Christoph Lampert, ECML, (2008))*

CCA often offers advantages above PCA. To illustrate, consider data samples available in two domains, X and Y, shown in Figure 2.6. In each of these samples, the data has higher variance in the noise direction than in the signal direction. As PCA concentrates on the variance of the data as the only criterion for projection directions, it will be sensitive to these high-noise directions and will not discover the meaningful representation along the signal direction (center depictions in Figure 2.6). CCA however can consider both X and Y domains simultaneously. Given that noise is unlikely to be correlated across data domains, CCA can identify that the data modalities are highly correlated when projected to the one (signal) direction, but yield little correlation when projected to the other (noise) direction. Thus, CCA ignores the semantically meaningless noise directions and instead chooses the directions of the signal in both the X and the Y domain (right depictions in Figure 2.6).

To derive CCA mathematically, we introduce *Pearson's correlation coefficient* (see Appendix) in the derivation of the 2-modality case of CCA. Let any paired data set, which we will denote X and Y, compose the paired data set $D = \{(x_1, y_1), \ldots, (x_n, y_n)\} \in \mathbb{R}^{d \times n}$. Pearson's correlation coefficient between this set is defined as:

$$\rho_{XY} = \frac{cov(X,Y)}{\sqrt{var(X)var(Y)}} \tag{2.32}$$

$$= \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - m_x)(y_i - m_y)}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - m_x)^2 \frac{1}{n}\sum_{i=1}^{n}(y_i - m_y)^2}}, \tag{2.33}$$

where m_x and m_y the empirical means of X and Y respectively. We assume that the data is centered at the origin (subtract the mean of the respective data set from each respective sample), from which it follows that the projected data is also centered at the origin ($\sum_i^n \hat{x}_i \longrightarrow \sum_i^n u_i^T x_i = u_i^T \sum_i^n x_i = 0$). Now we want to maximize the correlation between the *projections* of the data in each space, which is denoted e.g.

$$\hat{x}_i = u_x^T x_i \tag{2.34}$$

$$\hat{y}_i = u_y^T y_i. \tag{2.35}$$

So we introduce directions/vectors u_x and u_y with respect to which we can maximize the expression in Equation (2.32). This can be expressed with the following optimization problem:

$$\max_{u_x,u_y} \rho_{\hat{X}\hat{Y}} = \max_{u_x,u_y} \frac{\frac{1}{n}\sum_{i=1}^{n}\hat{x}_i\hat{y}_i}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\hat{x}_i^2\frac{1}{n}\sum_{i=1}^{n}\hat{y}_i^2}}, \tag{2.36}$$

and by expanding the projections, we arrive at the following expression:

$$\max_{u_x,u_y} \rho_{\hat{X}\hat{Y}} = \max_{u_x,u_y} \frac{\frac{1}{n}\sum_{i=1}^{n}u_{x_1}^T x_i u_{y_1}^T y_i}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(u_{x_1}x_i)^2\frac{1}{n}\sum_{i=1}^{n}(u_{y_1}y_i)^2}}. \tag{2.37}$$

By writing the sums over all samples in their matrix forms, Expression (2.37) can be *equivalently* expressed by *maximizing* the following objective function:

$$\textit{CCA objective function} \qquad \frac{u_x^T C_{XY} u_y}{\sqrt{(u_x^T C_{XX} u_x)(u_y^T C_{YY} u_y)}}. \tag{2.38}$$

where the the cross- and auto-covariance matrices of X and Y are denoted by C_{XY}, C_{XX}/C_{YY}, respectively. We can equivalently rewrite the objective function as the following constrained optimization problem:

$$u_x^T C_{XY} u_y \tag{2.39}$$

subject to:

$$u_x^T C_{XX} u_x = 1 \tag{2.40}$$

$$u_y^T C_{YX} u_y = 1 \tag{2.41}$$

We can rewrite this constrained optimization problem as an unconstrained function by using Lagrangian formalism as demonstrated previously with LDA in Section 2.2. The corresponding Lagrangian function is:

$$L(u_x, u_y, \mu, \nu) = u_x^T C_{XY} u_y - \mu(u_x^T C_{XX} u_x - 1) - \nu(u_y^T C_{YY} u_y - 1) \tag{2.42}$$

The minimization of this Lagrangian function is obtained as follows

$$\frac{\partial L}{\partial u_x}(u_x, u_y, \mu, \nu) = 0 \tag{2.43}$$
$$\Leftrightarrow C_{XY}u_y - 2\mu C_{XX}u_x = 0 \tag{2.44}$$
$$\Leftrightarrow C_{XY}u_y = 2\mu C_{XX}u_x \tag{2.45}$$

$$\frac{\partial L}{\partial u_y}(u_x, u_y, \mu, \nu) = 0 \tag{2.46}$$
$$\Leftrightarrow C_{YX}u_x - 2\nu C_{YY}u_y = 0 \tag{2.47}$$
$$\Leftrightarrow C_{YX}u_x = 2\nu C_{YY}u_y \tag{2.48}$$

Minimization of the CCA Lagrangian function results in following linear system:

$$C_{XY}u_y = 2\mu C_{XX}u_x \tag{2.49}$$
$$C_{YX}u_x = 2\nu C_{YY}u_y, \tag{2.50}$$

which we can write in matrix notation as follows

$$\begin{pmatrix} 0 & C_{XY} \\ C_{YX} & 0 \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} = \begin{pmatrix} (2\mu I & 0 \\ 0 & 2\nu I) \end{pmatrix} \begin{pmatrix} C_{XX} & 0 \\ 0 & C_{YY} \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix}. \tag{2.51}$$

Through algebraic simplification we can manipulate the above system in two ways. First, in order to make the matrix on the left side positive definite and allow it's optimization to be more efficient, we add the matrix and coefficients of the right side of both equations to the entire system. This results in the following linear system:

$$C_{XX}u_x + C_{XY}u_y = 2\mu C_{XX}u_x + C_{XX}u_x \tag{2.52}$$
$$C_{YX}u_x + C_{YY}u_y = 2\nu C_{YY}u_y + C_{YY}u_y, \tag{2.53}$$

which we write again in matrix notation

$$\begin{pmatrix} C_{XX} & C_{XY} \\ C_{YX} & C_{YY} \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} = \begin{pmatrix} 1 + 2\mu I & 0 \\ 0 & 1 + 2\nu I \end{pmatrix} \begin{pmatrix} C_{XX} & 0 \\ 0 & C_{YY} \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix}. \tag{2.54}$$

Second, we notice the symmetry of the cross-covariance matrices, thus

$$C_{XY} = C_{YX}, \tag{2.55}$$

which allows further algebraic simplification such that $1 + 2\mu = 1 + 2\nu$. Thus the system can be written as

$$\begin{pmatrix} C_{XX} & C_{XY} \\ C_{YX} & C_{YY} \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} = (1 + 2\mu) \begin{pmatrix} C_{XX} & 0 \\ 0 & C_{YY} \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix}. \tag{2.56}$$

Which we note is the form of a generalized eigenvalue problem. As such, we note that the eigenvalue $(1 + 2\mu)$ can be substituted by the variable λ, and the minimization problem is written as the following generalized eigenvalue problem:

$$\begin{pmatrix} C_{XX} & C_{XY} \\ C_{YX} & C_{YY} \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} = \lambda \begin{pmatrix} C_{XX} & 0 \\ 0 & C_{YY} \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix}. \tag{2.57}$$

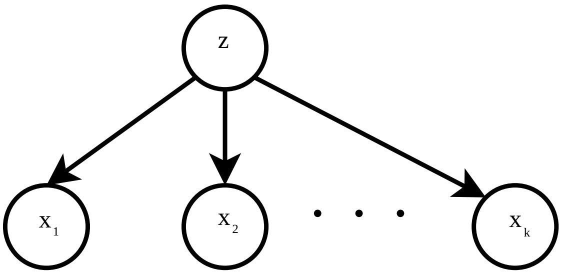
This solution of CCA in Equations (2.57) yields the first directions that maximize the correlation between the data projection in each space: u_x^1 and u_y^1. Maximization of the objective function in Expression (2.38) yields the correlation coefficient, explaining the magnitude of the correlation between the data when projected on u_x and u_y. The resulting vectors u_x^1 and u_y^1 are the first basis vectors of the lower-dimensional representations of X and Y; the assumption is that the lower-dimensional representation corresponds to the structure in the latent process Z. The next directions of maximum correlation between data projections can be discovered, similarly as in PCA and LDA, by sorting the remaining eigenvectors by their generalized eigenvalues, e.g. u_x^2 and u_y^2 with the next highest eigenvalues are the next directions.

2.3.1 Properties of CCA

This 2-modality derivation of CCA can be generalized to more than two modalities (illustrated in Figure 2.7), $\{X_1, \ldots, X_k\}$, by solving the following eigenvalue problem:

$$\begin{pmatrix} C_{11} & \ldots & C_{1k} \\ \vdots & \ddots & \vdots \\ C_{k1} & \ldots & C_{kk} \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_k \end{pmatrix} = \lambda \begin{pmatrix} C_{11} & \ldots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \ldots & C_{kk} \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_k \end{pmatrix}. \tag{2.58}$$

which includes 2-modality CCA as a special case.



(a) Multimodal data.

Figure 2.7: *With data available in multiple modalities, an underlying process may induce dependence between all observed output variables (modalities).*

This generalized solution in Equation (2.58) yields projection directions $u_1, \ldots, u_k$ that correspond to the maximum correlation between all of the spaces.

2.3.2 Special Cases of CCA

LDA. Note that after this derivation of the 2-modality case, when the Y modality consists of values in $\{-1, 1\}$ instead of real continuous values, CCA and LDA learn the same projection directions (making LDA a special case of CCA). (Cai et al. (2007))

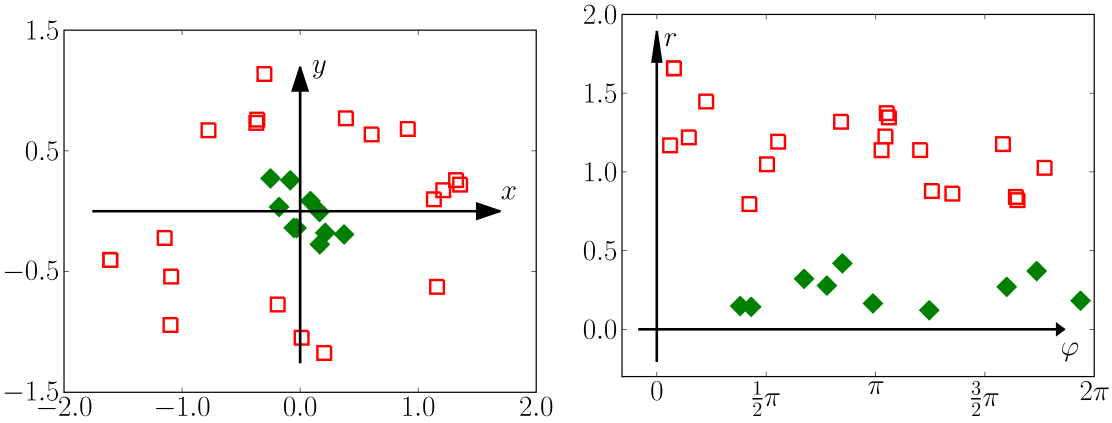
Least squares Regression. Least squares and CCA can be identically expressed if data are only available in two modalities, i.e. X and Y, and one modality contains the 1-dimensional labels corresponding to the other modality. (Sun et al. (2008) Sun et al. (2009))

Chapter 3

Nonlinear Dimensionality Reduction

When data is acquired from the natural world and represented digitally, it generally resides in very high-dimensional vector spaces, as described in Chapter 2. As we saw, by manipulating the basis vectors of the subspace in which the data is living, patterns which were previously non-discernible can be made reasonably salient. As illustrated by the linear methods of PCA, LDA, and CCA, this is commonly done by projecting the entire data set in question to a ”more appropriate” subspace of fewer dimensions (dimensionality reduction), in which the previously unidentifiable structure could be discerned. However, when the data contain more complex patterns, these linear methods offer little help in unveiling them. Consider for example, the data in Figure 3.1 (a): they exhibit a clear ”ring-like” pattern consisting of samples in two classes, red squares and green squares. But yet, there is no linear projection that leads to a lower dimensional representation in which the classes are well separated (every linear projection would heavily mix the classes).

Consider the same sample data in Figure 3.1 (a). There may be a more suitable coordinate system than the Cartesian to which the data could be mapped *before* being able to apply LDA to find a projection direction upon which the two classes are distinguished, shown in (b) of the figure (e.g. a projection direction parallel to the r-axis would suffice). As such, we need some mapping function to perform this coordinate transform where the important descriptions of the data classes, the *features*, are now linearly separable in this new space. We call the original space, which in this example is the Cartesian coordinates, the *input space*, and the space the data is mapped to, i.e. that of the polar coordinates the *feature space*. We call this mapping function ϕ the *feature map*. The specific feature map ϕ for the polar coordinates of $(x, y) \in \mathbb{R}^2$, $(x, y) \mapsto (r, \varphi)$ is expressed by:



(a) Cartesian coordinates. (b) Polar coordinates.

Figure 3.1: *Mapping from Cartesian coordinates to Polar coordinates, $(x, y) \mapsto (\varphi, r)$. The pattern in the data is complex and difficult to characterize when represented in Cartesian coordinates (a), but when transformed to polar coordinates, the structure becomes simpler to characterize e.g. via Linear Discriminant Analysis, the two classes within the data could be easily separated (b). (Figures from Lampert (2009)).*

$$\phi : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} \sqrt{x^2 + y^2} \\ arctan\frac{y}{x} \end{pmatrix}. \tag{3.1}$$

Thus by mapping input data (x, y) from its original space to a feature space by said nonlinear map ϕ, the complex nonlinear pattern in the data is now much more clear for the application of linear dimensionality reduction techniques to further extract the pattern of the two classes in the data.

The methods from the previous chapter can be extended and generalized to account for such nonlinear patterns. In order to identify patterns of greater complexity in a data sample, there needs to be a similar intermediate steps in order for the previously discussed methods to be able to detect them. Namely, a more suitable representation of the data for the task at hand: a coordinate transformation before dimensionality reduction.

This general approach can be applied to any input data set $x_1, \ldots, x_n \in \mathcal{X}$ to map it to its individual ϕ-induced feature space $\mathcal{H}$ as follows:

$$\phi : \mathcal{X} \to \mathcal{H} \tag{3.2}$$

$$x \mapsto \phi(x), \tag{3.3}$$

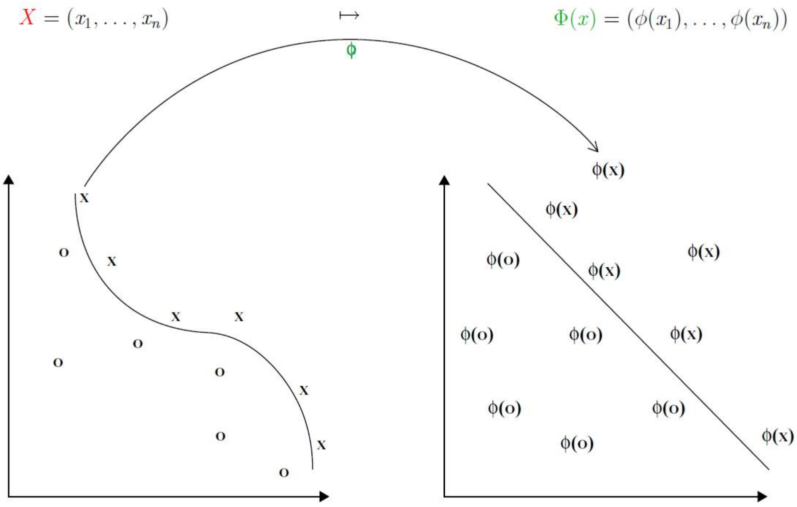


Figure 3.2: *nonlinear feature mapping. The feature map, transformation function ϕ, takes a nonlinear pattern in the original vector space and embeds it in another space, in which the pattern becomes linear. (Figure courtesy of Schölkopf and Smola (2002))*

which aids in representing data containing nonlinearities. See Figure 3.2 for a general illustration.

In this chapter, I will describe how the dimensionality reduction techniques of Chapter 2 can be generalized to account for nonlinearities in data. As in the previous chapter, the nonlinear extensions can be mathematically expressed as generalizations of Principle Components Analysis (PCA), thus we will again begin there.

3.1 PCA in Feature Space

We will outline the above concepts in the context of PCA. Recall that PCA reduces the dimensionality of a data set by finding the direction(s) in the data where the data projections have maximal variance by calculating the top eigenvectors of the covariance matrix of the data set (e.g. the principle components). All of this can be performed in feature space in order to find the potentially nonlinear principle components.

Let $X = (x_1, \ldots, x_n) \in \mathbb{R}^{d \times n}$ be the data set centered at the origin, the covariance matrix of which is $C_{XX} = \frac{1}{n}XX^T$. PCA, before being mapped to a feature space, is expressed by the optimization problem:

$$\max_{u} \quad u^T C_{XX} u, \tag{3.4}$$

subject to:

$$u^T u = 1 \tag{3.5}$$

Now we map X to some feature space $\mathcal{H}$ by the feature map ϕ. We perform this mapping as follows:

$$X = (x_1, \dots, x_n) \mapsto \Phi(x) = (\phi(x_1), \dots, \phi(x_n)), \tag{3.6}$$

calling the result Φ, which is the representation of X in $\mathcal{H}$ as mapped by ϕ.

Now the covariance matrix of X in the feature space $\mathcal{H}$ becomes

$$C_{\Phi\Phi} := \frac{1}{n} \sum_{i=1}^{n} \phi(x_i)\phi(x_i)^T. \tag{3.7}$$

Substituting the $C_{\Phi\Phi}$ into the previous PCA Expression (3.4) we arrive at the following optimization problem:

$$\max_{\tilde{u}} \quad \tilde{u}^T C_{\Phi\Phi} \tilde{u}, \tag{3.8}$$

subject to:

$$\tilde{u}^T \tilde{u} = 1, \tag{3.9}$$

the solution for $\tilde{u}$ of which can be obtained by solving the following eigenvalue problem:

$$C_{\Phi\Phi} \tilde{u} = \lambda \tilde{u}, \tag{3.10}$$

where the eigenvector $\tilde{u}_i$ with the highest eigenvalue λ_i is the first principle component within the feature space $\mathcal{H}$, analogous to the eigenvector u_i from PCA performed in the input space (Section 2.1). As in the original PCA, the further principle components can be obtained by taking the next $\tilde{u}_i$ with the highest λ_i. By mapping the input data X first to a feature space $\mathcal{H}$ via ϕ, this is a pre-processing step to 'linearize' the nonlinear principle components, such that PCA can locate the correct directions to which they correspond. From the eigenvalue problem in Equation (3.10) we see that the solution $\tilde{u}$ lies in the span of the training examples $\phi(x_1), \ldots, \phi(x_n)$, and thus the numerical solution can be expressed as:

$$\tilde{u} = \sum_{i}^{n} \alpha_i \phi(x_i) \tag{3.11}$$

3.2 Implicit Mapping into Feature Space

We use the notation Φ to denote the matrix where each row corresponds to a feature mapped data point, $\phi(x_i)^T$. Because $\tilde{u}^T C_{\Phi\Phi} \tilde{u} = \tilde{u}^T \Phi^T \Phi \tilde{u}$ and $\tilde{u} = \Phi\alpha$ (Equation (3.11)), we can take the optimization of PCA in Expression (3.8) and by searching over $\alpha_1, \ldots, \alpha_n$, we can express the equivalent problem:

$$\max_{\alpha} \quad \alpha^T \Phi^T \Phi \Phi^T \Phi \alpha, \tag{3.12}$$

subject to:

$$\tilde{u}^T \tilde{u} = 1 \tag{3.13}$$

$$\implies \sum_{i,j=1}^{n} \alpha_i \alpha_j \langle \phi(x_i), \phi(x_j) \rangle = 1. \tag{3.14}$$

We note that ϕ is only present in the form of inner products, e.g., $\phi^T \phi = \langle \phi, \phi \rangle$. We can define the *kernel function* $k : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ of ϕ by the following identity:

$$k(x, x') := \langle \phi(x), \phi(x') \rangle. \tag{3.15}$$

Every k of this type is *symmetric*.

Proof. Let $k(x, x') = \langle \phi(x), \phi(x') \rangle$. Show that

$$\forall x, x' \in \mathcal{X} : k(x', x) = k(x, x'). \tag{3.16}$$

For arbitrary $x, x' \in \mathcal{X}$:

$$k(x, x') = \langle \phi(x), \phi(x') \rangle = \langle \phi(x'), \phi(x) \rangle = k(x', x) \tag{3.17}$$

□

Thus, k is symmetric because the inner product $\langle \cdot, \cdot \rangle$ is symmetric.

Every k is also *positive definite*. That is, for all $c_1, \dots, c_n \in \mathbb{R}$, all $N \in \mathbb{N}$, and for all $x_1, \dots, x_N$, the matrix $K \in \mathbb{R}^{N \times N}$ where $K_{ij} = (k(x_i, x_j))_{i,j}$ is positive semidefinite.

Proof. Let $c_1, \dots, c_n$ be arbitrary $\in \mathbb{R}$:

$$\begin{aligned}
\sum_{i,j=1}^{n} c_i K_{ij} c_j &= \sum_{i=1,j=1}^{n} c_i k(x_i, x_j) c_j && (3.18)\\
&= \sum_{i=1,j=1}^{n} c_i \langle \phi(x_i), \phi(x_j) \rangle c_j && (3.19)\\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} \langle c_i \phi(x_i), \phi(x_j) c_j \rangle && (3.20)\\
&= \langle \sum_{i=1}^{n} c_i \phi(x_i), \sum_{j=1}^{n} \phi(x_j) c_j \rangle && (3.21)\\
&= \| \sum_{i=1}^{n} c_i \phi(x_i) \|^2 \geq 0 && (3.22)
\end{aligned}$$

□

Thus ϕ induces a positive definite kernel function k. Furthermore, for the kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ it holds that for any finite set of data points $x_1, \dots, x_n \in \mathcal{X}$, there exists a *kernel matrix* K

$$K_{ij} := k(x_i, x_j) \tag{3.23}$$

which is *symmetric* ($K_{ij} = K_{ji}$) as well as *positive semidefinite* (e.g., for all vectors $t \in \mathbb{R}^n$ it holds that $\sum_{i,j=1}^{n} t_i K_{ij} t_j \geq 0$). This is also called a *Gram matrix* of k. Additionally, it can be proved (see Schölkopf and Smola (2002)) that for any such positive definite k, there exists a Hilbert space $\mathcal{H}$ and a feature map $\phi : \mathcal{X} \mapsto \mathcal{H}$ such that

$$k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}} \tag{3.24}$$

where the inner product in $\mathcal{H}$ is denoted by $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. This is called *Mercer's condition.*

The kernel k provides all the necessary information about the ϕ mapping of X. One can substitute k into any algorithm that only needs inner products of the data points, and thereby do not need the explicit mapping of the data points with ϕ. This is referred to as the *kernel trick* (Schölkopf and Smola (2002)).

To illustrate this, we consider an optimization problem with the kernelization method of ϕ; specifically we evaluate the problem of PCA in feature space (Equation (3.4)). Solving this optimization problem yields $\alpha_1, \ldots, \alpha_i \in \mathbb{R}$ for which we can compute the projection (solution shown in Equation (3.11)):

$$\tilde{u} = \sum_i^n \alpha_i \phi(x_i). \tag{3.25}$$

This solution for $\tilde{u}$ is computable without ϕ, given

$$\begin{aligned} \tilde{x}_i &= \tilde{u}_i^T x & (3.26)\\ &= \langle \tilde{u}, \phi(x) \rangle & (3.27)\\ &= \langle \sum_i^n \alpha_i \phi(x_i), \phi(x) \rangle & (3.28)\\ &= \sum_i^n \alpha_i \langle \phi(x_i), \phi(x) \rangle & (3.29)\\ &= \sum_i^n \alpha_i k(x_i, x) & (3.30) \end{aligned}$$

This demonstrates that we only need the inner product values from the kernel function k to evaluate and derive solutions to dimensionality reduction problems such as PCA (e.g. with function form of dimensionality reduction as $f(x) = \tilde{x}$ where $f : \mathcal{X} \mapsto \mathbb{R}^n$), and thus we replace ϕ with k. We call this process *kernelization.*

As shown, the whole kernelization process needs only the inner products (kernel function

k), and not the mapping (non)linear function ϕ itself (kernel trick). This means we can do (non)linear dimensionality reduction with any input for which we can define a positive definite kernel function. Kernelization has several useful implications. First, needing only the inner product information contained in the kernel function simplifies costly demands of the problem. As we do not need the explicit knowledge of ϕ in order to learn in its induced feature space $\mathcal{H}$, and by not having to compute and store ϕ, we can save a lot of space and computation time. As shown above, we can acquire the numerical solution for any projection in this implicitly defined feature space based only on the inner products of the data points. Second, it allows for a generalized solution: the previous linear dimensionality reduction methods are generalized to account for nonlinear data patterns. In this way, kernelization is also used as a pre-processing step for the dimensional reduction of data containing nonlinear structure: the data is nonlinearly mapped into some new feature space, in which the complex patterns are now handleable linearly as described in Chapter 2.

3.3 Kernel Canonical Correlation Analysis (KCCA)

Although all of the dimensionality reduction techniques can be nonlinearly extended via kernelization as described above, we will focus on the kernelization of CCA (KCCA) (see e.g. Leurgans et al. (1993); Lai and Fyfe (2000); Bach and Jordan (2002); Hardoon et al. (2004)).

CCA (Section 2.3) seeks to maximize the correlation between principle directions in two or more separate data domains, or modalities of some underlying process, simultaneously. Let $X = \{x_1, \dots, x_n\}$ and $Y = \{y_1, \dots, y_n\}$ be two modalities and u_x and u_y be the projection directions within the domains, respectively. CCA can be expressed as the following optimization problem:

$$\max_{u_x, u_y} \frac{u_x^T C_{XY} u_y}{\sqrt{(u_x^T C_{XX} u_x)(u_y^T C_{YY} u_y)}} = \max_{u_x, u_y} \frac{u_x^T X^T Y u_y}{\sqrt{(u_x^T X^T X u_x)(u_y^T Y^T Y u_y)}} \tag{3.31}$$

which can be solved as a generalized eigenvalue problem (Section 2.3), yielding the first directions u_x^1 and u_y^1 of maximum correlation between the projected X and Y data sets.

As described previously in Section 3.2, in order to kernelize the above expression of CCA, we need to define a mapping of the data sets

$$X = \{x_1, \dots, x_n\} \tag{3.32}$$
$$Y = \{y_1, \dots, y_n\} \tag{3.33}$$

into their respective kernel Hilbert spaces, as denoted by the kernel functions

$$k_x : \mathcal{X} \times \mathcal{X} \to \mathbb{R} \tag{3.34}$$
$$k_y : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R} \tag{3.35}$$

with induced feature mapping functions

$$\Phi_x : \mathcal{X} \to \mathcal{H}_x \tag{3.36}$$
$$\Phi_y : \mathcal{Y} \to \mathcal{H}_y \tag{3.37}$$
$$\Phi_x = (\phi_x(x_1), \dots, \phi_x(x_n)) \tag{3.38}$$
$$\Phi_y = (\phi_y(y_1), \dots, \phi_y(y_n)) \tag{3.39}$$

which, when we substitute these ϕ-mapped input data sets in the previous expression, yields the following expression of CCA within the kernel Hilbert space $\mathcal{H}$:

$$\max_{\tilde{u}_x, \tilde{u}_y} \quad \frac{\tilde{u}_x^T \Phi_x^T \Phi_y \tilde{u}_y}{\sqrt{(\tilde{u}_x^T \Phi_x^T \Phi_x \tilde{u}_x)(\tilde{u}_y^T \Phi_y^T \Phi_y \tilde{u}_y)}}. \tag{3.40}$$

We can denote the projection directions $\tilde{u}_x$ and $\tilde{u}_x$ as the linear combination between coefficients within the Hilbert spaces and with respective mapped data sets. This results in the dual representation of the projection directions within the Hilbert spaces:

$$\tilde{u}_x = \sum_i \alpha_i \phi_x(x_i) = \Phi_x \alpha \tag{3.41}$$
$$\tilde{u}_y = \sum_i \beta_i \phi_y(y_i) = \Phi_y \beta. \tag{3.42}$$

Substituting these dual variables into the previous optimization problem in Expression (3.40), we arrive at its dual representation (as with kernel PCA and as illustrated in Section 3.2):

$$\max_{\alpha,\beta} \quad \frac{\alpha^T\Phi_x^T\Phi_x\Phi_y^T\Phi_y\beta}{\sqrt{(\alpha^T\Phi_x^T\Phi_x\Phi_x^T\Phi_x\alpha)(\beta^T\Phi_y^T\Phi_y\Phi_y^T\Phi_y\beta)}}. \tag{3.43}$$

Furthermore, we denote the kernel matrices as

$$[K_x]_{ij} := k_x(x_i, x_j) = \langle \phi_x(x_i), \phi_x(x_j)\rangle, \tag{3.44}$$

$$[K_y]_{ij} := k(y_i, y_j) = \langle \phi(y_i), \phi(y_j)\rangle, \tag{3.45}$$

$$K_x := \Phi_x^T\Phi_x, \tag{3.46}$$

$$K_y := \Phi_y^T\Phi_y. \tag{3.47}$$

and substituting these into Expression (3.43) results in the following kernelized objective function to be maximized:

$$\textit{KCCA objective function} \qquad \frac{\alpha^T K_x K_y\beta}{\sqrt{(\alpha^T K_x K_x\alpha)(\beta^T K_y K_y\beta)}}. \tag{3.48}$$

Analogous as performed by CCA in the previous chapter, we can write the above expression as the following constrained optimization problem:

$$\max_{\alpha,\beta} \alpha^T K_x K_y \beta \tag{3.49}$$

subject to:

$$\tag{3.50}$$

$$\alpha^T K_x K_x \alpha = 1 \tag{3.51}$$

$$\beta^T K_y K_y \beta = 1, \tag{3.52}$$

upon which we can utilize Lagrangian formalism to construct the following unconstrained minimization problem:

$$L(\lambda_\alpha, \lambda_\beta, \alpha, \beta) = \alpha^T K_x K_y \beta - \lambda_\alpha(\alpha^T K_x K_x \alpha - 1) - \lambda_\beta(\beta^T K_y K_y \beta - 1). \tag{3.53}$$

The minimization of this Lagrangian function is obtained as follows

$$\begin{aligned} \frac{\partial L}{\partial \lambda_\alpha}(\lambda_\alpha, \lambda_\beta, \alpha, \beta) = 0 && (3.54) \\ \Leftrightarrow K_x K_y \beta - 2\lambda_\alpha K_x K_x \alpha = 0 && (3.55) \\ \Leftrightarrow K_x K_y \beta = 2\lambda_\alpha K_x K_x \alpha && (3.56) \\ && (3.57) \end{aligned}$$

$$\begin{aligned} \frac{\partial L}{\partial \lambda_\beta}(\lambda_\alpha, \lambda_\beta, \alpha, \beta) = 0 && (3.58) \\ \Leftrightarrow K_y K_x \alpha - 2\lambda_\beta K_y K_y \beta = 0 && (3.59) \\ \Leftrightarrow K_y K_x \alpha = 2\lambda_\beta K_y K_y \beta && (3.60) \end{aligned}$$

from which, following the same steps as with CCA in Chapter 2 (again, algebraically manipulating the system such that the matrix on the left is positive definite), we can derive the following generalized eigenvalue problem:

$$\begin{pmatrix} K_x K_x & K_x K_y \\ K_y K_x & K_y K_y \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \lambda \begin{pmatrix} K_x K_x & 0 \\ 0 & K_y K_y \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}. \qquad (3.61)$$

The solution to the above eigenvalue problem yields the coefficients α and β in the Hilbert spaces $\mathcal{H}_x$ and $\mathcal{H}_y$, from which we obtain the directions $\tilde{u}_x$ and $\tilde{u}_x$ by Equations (3.41) and (3.42), respectively, that maximize the correlation between the projections of $\tilde{X}$ and $\tilde{Y}$. We can again use the Representer theorem to acquire a numerical solution by rewriting this dual representation of the projections (the result from the first use of the Representer theorem, shown in Equations (3.41) and (3.42)) into the their primal form, yielding weight vectors $\tilde{u}_x$ and $\tilde{u}_y$. These vectors in the Hilbert spaces are analogous to the u_x and u_y in the input spaces resulting from Equation (2.57).

3.3.1 Tikhonov Regularization

In the case that either K_x or K_y is invertible, the solution to Equation (2.57) will yield perfect correlation, but is trivial and does not learn anything (Leurgans et al. (1993); Bach and Jordan (2002); Hardoon et al. (2004)). This type of degenerate solution requires additional steps to be avoided.

As discussed in Chapter 1, we need to regularize solutions of optimization problems in order to avoid overfitting to the given data set and not being able to make general statements about unseen data. In other words, if we only minimize the training error (Chapter 1, Figure 1.2), this can lead to numerical instabilities and poor generalization ability. Thus far for dimensionality reduction we have considered functions of the form in Equation (3.65), but with the regularization term set to zero.

Consider the optimization problem for KCCA in Expression (3.49) in the case of either an invertible K_x or K_y. As α and β are directions in their respective Hilbert spaces $\mathcal{H}_x$ and $\mathcal{H}_y$, we can arbitrarily fix e.g. α, and look for a maximally correlated direction in the other space. Let α be fixed and set $\beta := K_y^{-1} K_x \alpha$. Substituting these values into Expression (3.49), KCCA is expressed with the optimization problem:

$$\alpha^T K_x K_y K_y^{-1} K_x \alpha \Leftrightarrow \alpha^T K_x K_x \alpha \tag{3.62}$$

subject to:

$$\alpha^T K_x K_x \alpha = 1 \tag{3.63}$$

$$(K_y^{-1} K_x \alpha)^T K_y K_y K_y^{-1} K_x \alpha \Leftrightarrow \alpha^T K_x K_x \alpha = 1. \tag{3.64}$$

From this illustration we can observe the problem with invertible kernel matrices. The expression we want to maximize (Expression (3.62)) is identical to its constraints set to 1. This is a trivial solution because e.g. once we arbitrarily set direction α, we can find a direction β with which it is perfectly correlated ($\rho = 1$).

Therefore we need to enforce the learning of non-trivial directions through regularization. Because in the situation of learning trivial directions that yield perfect correlation there is minimal covariance in the solution, we want to enforce some degree of covariance to exist in the solution (in order that the matrices are no longer invertible and e.g. $K_y K_y^{-1} = I$). Additionally we want to control overfitting of the solution to the training data, and thus need to control the flexibility of the learnable projection directions in the Hilbert space by introducing a regularization term to enforce smooth gradients in the associated projection functions. *Tikhonov regularization* enforces that the projection function $f_{proj}(\tilde{x}) = \tilde{u_x}^T \Phi_x = \alpha^T \Phi_x^T \Phi_x = \alpha^T K_x$ will have a smooth gradient. This involves the addition of a regularization term to penalize the solution via the L_2-norm, specifically by punishing large $\|\nabla f_{proj}(\tilde{x})\|^2$ to some parameter-controlled degree (regularization parameter), and as such the norm of the projected solution $\|\tilde{u}_x\|_2$ will be accordingly punished (small). As such, the learned solution is more generalizable to unseen (test) data as large oscillations in the solution (which would i.e. perfectly match the training data and produce high error with testing data) are discouraged with the regularization term. The resulting solution is, in other words, smooth with respect to the ambient space.

Thus we introduce Tikhonov regularization to the objective function for KCCA in Equation (3.48) to avoid degenerate solutions and discourage overfitting, resulting in the following expression:

$$\textit{KCCA objective function} \qquad \frac{\alpha^T K_x K_y \beta}{\sqrt{(\alpha^T K_x K_x \alpha + \varepsilon_x \|\tilde{u}_x\|^2)(\beta^T K_y K_y \beta + \varepsilon_y \|\tilde{u}_y\|^2)}}, \tag{3.65}$$

where ε_x and ε_y are the regularization parameters controlling the degree of penalization (regularization) of the norms in each Hilbert space. Since $\|\tilde{u}_x\|^2 = \tilde{u}_x^T \tilde{u}_x = \alpha^T \Phi_x^T \Phi_x \alpha = \alpha^T K_x \alpha$, this can be equivalently rewritten as

$$\textit{KCCA objective function} \qquad \frac{\alpha^T K_x K_y \beta}{\sqrt{(\alpha^T K_x K_x \alpha + \varepsilon_x \alpha^T K_x \alpha)(\beta^T (K_y K_y \beta + \varepsilon_y \beta^T K_y \beta)}}, \tag{3.66}$$

and simplified as the following expression

$$\textit{KCCA objective function} \qquad \frac{\alpha^T K_x K_y \beta}{\sqrt{\alpha^T (K_x K_x + \varepsilon_x K_x) \alpha \beta^T (K_y K_y + \varepsilon_y K_y) \beta}}. \tag{3.67}$$

With the identical steps as described with the unregularized version of KCCA, we can write the above regularized expression as a constrained optimization problem, where both parts of the denominator must be equal to 1, and we maximize

$$\alpha^T K_x K_y \beta \tag{3.68}$$

subject to:

$$\alpha^T K_x K_x \alpha + \varepsilon_x \alpha^T K_x \alpha = 1 \tag{3.69}$$

$$\beta^T K_y K_y \beta + \varepsilon_y \beta^T K_y \beta = 1, \tag{3.70}$$

with the corresponding Lagrangian function:

$$L(\lambda_\alpha, \lambda_\beta, \alpha, \beta) = \alpha^T K_x K_y \beta - \lambda_\alpha(\alpha^T K_x K_x \alpha + \varepsilon_x \alpha^T K_x \alpha - 1) - \lambda_\beta(\beta^T(K_y K_y \beta + \varepsilon_y \beta^T K_y \beta - 1). \tag{3.71}$$

The minimization of this Lagrangian function is obtained as follows

$$\frac{\partial L}{\partial \lambda_\alpha}(\lambda_\alpha, \lambda_\beta, \alpha, \beta) = 0 \tag{3.72}$$

$$\Leftrightarrow K_x K_y \beta = 2\lambda_\alpha(K_x K_x \alpha + \varepsilon_x K_x \alpha) \tag{3.73}$$

$$\tag{3.74}$$

$$\frac{\partial L}{\partial \lambda_\beta}(\lambda_\alpha, \lambda_\beta, \alpha, \beta) = 0 \tag{3.75}$$

$$\Leftrightarrow K_y K_x \alpha = 2\lambda_\beta(K_y K_y \beta + \varepsilon_y K_y \beta) \tag{3.76}$$

from which, by following the same algebraic steps with regular KCCA, we can derive the following generalized eigenvalue problem:

$$\begin{pmatrix} K_x(K_x + \varepsilon_x I) & K_x K_y \\ K_y K_x & K_y(K_y + \varepsilon_y I) \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \lambda \begin{pmatrix} K_x(K_x + \varepsilon_x I) & 0 \\ 0 & K_y(K_y + \varepsilon_y I) \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}. \tag{3.77}$$

As with KCCA described earlier, the solution to the above eigenvalue problem yields the coefficient vectors α^1 and β^1, which maximize the correlation between the Φ_X and Φ_Y data projections. We acquire a numerical solution for the first weight vectors $\tilde{u}_x^1$ and $\tilde{u}_y^1$ by Equations (3.41) and (3.42). These vectors in the Hilbert spaces are analogous to the u_x^1 and u_y^1 in the input spaces resulting from Equation (2.57). As with the linear dimensionality reduction methods, to obtain the further directions of maximal correlation, we simply take the next eigenvectors α^i and β^i corresponding to the next highest eigenvalues λ^i, leading to weight vector solutions of $\tilde{u}_x^i$ and $\tilde{u}_y^i$.

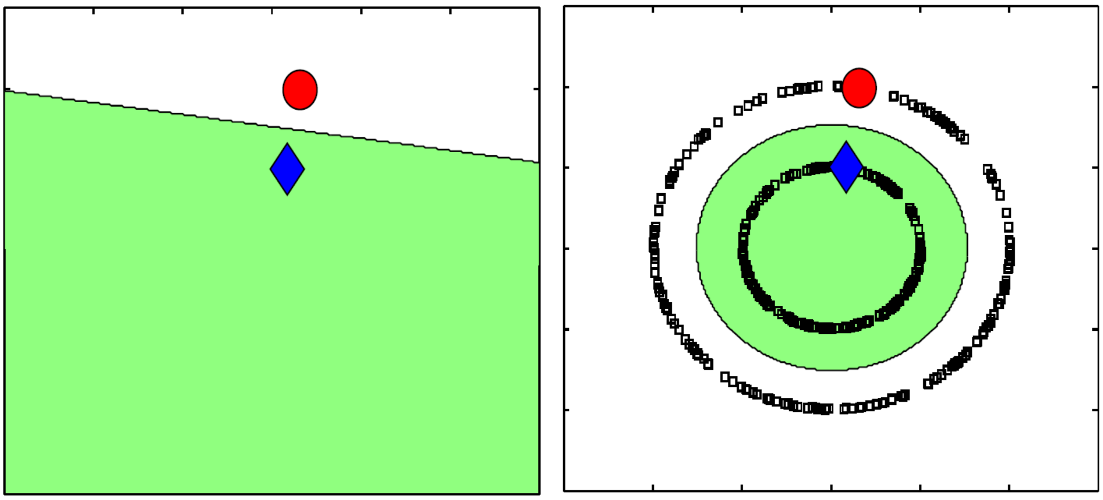
Chapter 4

Semi-supervised Learning

Until now, we have observed linear and nonlinear dimensionality reduction techniques under the unsupervised (e.g. PCA) and the supervised (e.g. LDA, CCA) learning frameworks. Now we will discuss a learning framework which can incorporate methods from both of these frameworks, called *semi-supervised learning*.

In the introduction to machine learning in Chapter 1, we discussed the different types of learning methods and the variety of data from which they can learn. In supervised learning methods, all data samples under consideration have corresponding labels, indicating some quality of interest for each data point (i.e. indication of class to which the data points belong), whereas in unsupervised learning, we learn non-parametrically from data without any corresponding labels. Semi-supervised learning can utilize a mix of data samples: those with corresponding labels and those without. These methods can be very powerful as an effect of the larger data sets they can consider and have the benefit of being less expensive in terms of requiring less sample labels. However, there is a much greater demand on the assumptions of the model. These strong model assumptions thus require the rigorous selection of the model and accordingly, the structuring of this model to the problem is of much greater importance. However when we select a good model for the problem at hand and can make use of the unlabeled data, we can achieve greater inferential performance. See Belkin et al. (2006), Chapelle et al. (2006), von Luxburg (2007), Zhou and Schölkopf (2006), and Zhu (2005).

Often, this inclusion of additional (unlabeled) data can restructure our original hypotheses about the patterns in a data set. For example, our original hypothesis about a data could lead to the learning of a discriminant function, with which to project the data in which the classes are maximally linearly separable (i.e. LDA in Section 2.2). However, when we include unlabeled samples with our labeled samples in order to learn the discriminant function (e.g. semi-supervised discriminant analysis, see Cai et al. (2007)), we may see



(a) Small sample size

(b) Larger sample size

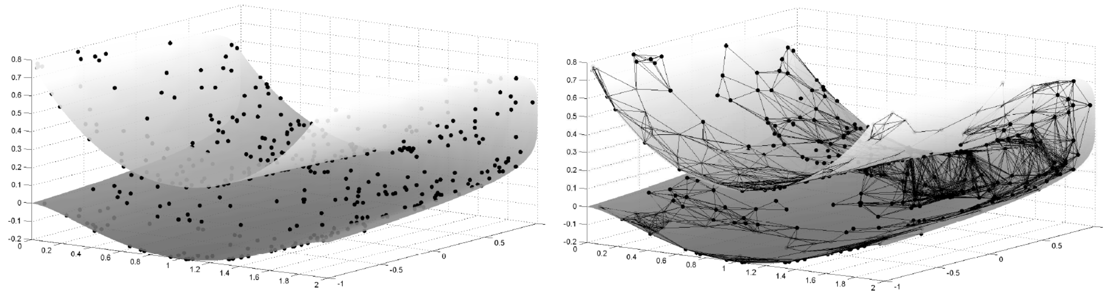
Figure 4.1: *With a small labeled sample size, the data exhibit e.g. a pattern which could be projected such that the circle and diamond are discriminated (a), but with the inclusion of unlabeled data in a semi-supervised framework, we are forced to readjust our hypotheses, as the data now exhibit this circular pattern instead of the hypothesized linearly separable one (b). (Figures from Belkin et al. (2006)).*

that the relationship of the classes is not linear, but rather reveals a different structure (Belkin et al. (2006)). See Figure for illustration.

4.1 Graph-based Methods: Manifold Regularization

Most semi-supervised methods utilize the geometry of the underlying distribution of the observed data samples, and exploit its properties to enable the use of unlabeled data. This forms a class of methods called *manifold regularization*, where a regularization term is formed from labeled and unlabeled samples and is included in a given optimization problem, thereby making the problem semi-supervised. Here we briefly introduce the idea behind this type of regularization.

In order to make use of the geometry of the distribution, which we cannot directly observe, we first need to form some estimate of it based on the data samples from it that we do have. First, we make the *manifold assumption*, which states that high-dimensional data lies on a low-dimensional manifold structure $\mathcal{M}$, with $\mathcal{M} \subset \mathbb{R}$ (Belkin et al., 2006). A manifold is an n-dimensional space that locally looks like n-dimensional Euclidean space, but whose global structure may be non-Euclidean. The manifold assumption for high-dimensional data is the basis of most forms of semi-supervised learning methods.

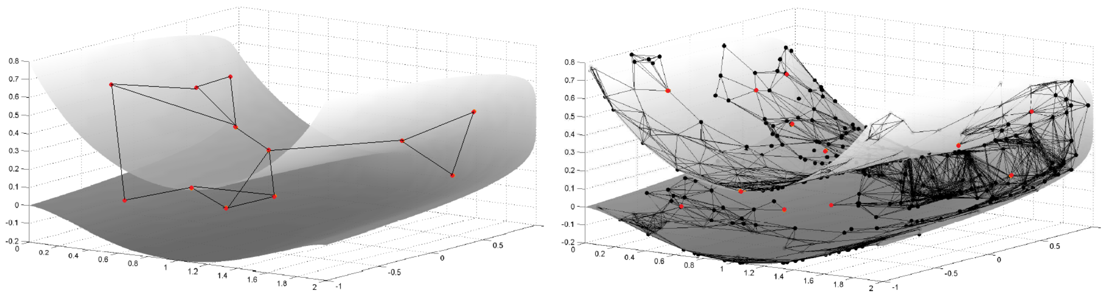


(a) Samples of manifold (b) Graph estimate of manifold

Figure 4.2: *Example of samples from a low-dimensional manifold (a), and the corresponding graph estimate based on the samples of the manifold (b). (Figures from von Luxburg, MLSS, Bousquet et al. (2004)).*

With this assumption, we can non-parametrically estimate the geometric structure of the data manifold using the data samples. This is done by defining a graph whose nodes consist of the data points and connecting edges between the data points that indicate the ”weight” or ”similarity” between those data points. See Figure 4.2 for an illustration of observed data points and the manifold structure on which they lie, and a graph estimate of that manifold structure.

Estimating the manifold structure does not require sample labels. We are able to make use of data samples with and without labels and thus define a graph whose nodes consist of samples from both labeled and unlabeled datasets (e.g. Hein et al. (2006); von Luxburg (2007)) In Section 3.3.1 we introduced Tikhonov regularization as a way to avoid degenerate solutions with KCCA and as a method to reduce overfitting of the learned solution and increase its generalizability to unseen data. Manifold regularization methods learn a more generalizable solution (when the manifold assumption holds with the given dataset), but they can do so by making use of the mixed-data graph estimated manifold. This type of regularization regularizes the learned solution with respect to the estimated manifold structure, such that the solution is smooth along the manifold. In doing so, these methods all estimate a function $f_{\mathcal{M}}$ along the mixed-data graph estimate of the manifold, and encourage solutions to have a small gradient with respect to this estimated structure. Thereby when the manifold assumption holds with a given dataset, not only do these methods allows one to make use of the data without labels, but also allow for exploitation of a larger sample size and correspondingly provide better estimate of the underlying manifold structure. When these conditions are met and we can obtain a decent estimate of the manifold’s structure, these methods offer greater generalizability and reduction in overfitting. See Figure 4.3 for an illustration of this concept, which also again demonstrates an example of needing to restructure hypotheses about the data, as illustrated previously in Figure 4.



(a) Poor estimate

(b) Better estimate

Figure 4.3: *The number of data samples of a manifold corresponds to how well the structure of the manifold is estimated. The manifold estimated by a graph consisting of few data points is a poor estimate of the actual structure of the data manifold (a). However, with more data points, the graph forms a better estimate of the manifold's structure. (Figures from von Luxburg, MLSS, Bousquet et al. (2004)).*

As we are particularly interested in Laplacian regularization, we will only discuss the derivation of the graph Laplacian and its associated regularization.

4.1.1 Laplacian Regularization

Laplacian regularization forms a graph estimate of the manifold structure called a *graph Laplacian*, and regularizes the solution of the optimization problem with respect to it.

Suppose we have the data set $\dot{X} = \{x_1, \ldots, x_p\}$ which is a composite set of the data set with known labels, $X = \{x_1, \ldots, x_n\}$ (i.e. of the supervised training data set $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, and the data set without known labels, $\tilde{X} = \{x_{n+1}, \ldots, x_p\}$. As we want to estimate the manifold structure $\mathcal{M}$ of dataset $\dot{X}$, we want estimate the gradient of a function f along $\mathcal{M}$. We thus begin the graph estimate of $\mathcal{M}$ by estimating the gradient of the function f_{proj} that projects the data onto the manifold

$$\nabla f_{proj}(x_i, x_j) = \frac{f(x_i) - f(x_j)}{x_i - x_j}. \tag{4.1}$$

As we want to use the manifold estimate in Equation (4.1) for regularization, f_{proj} should be smooth, and accordingly $||\nabla f_{proj}||^2$ should be small (Belkin et al. (2006)). Thus we define the following regularization functional

$$\frac{1}{2}\sum_{i,j}^{n,n} w_{ij}\left(f(x_i)-f(x_j)\right)^2 = \frac{1}{2}\sum_{i,j}^{n,n} w_{ij}\left(f(x_i)^2 - 2f(x_i)f(x_j) + f(x_j)^2\right) \tag{4.2}$$

$$= \sum_{i,j}^{n,n} w_{ij} f(x_i)^2 - \sum_{i,j}^{n,n} w_{ij}\left(f(x_i)f(x_j)\right) \tag{4.3}$$

$$= f_j\left(\sum_{j}^{n} w_{ij} - \sum_{i,j}^{n,n} w_{ij}\right) f_i \tag{4.4}$$

which pays attention to the similarity between data points (weights) encoded in w_{ij}. This can be further simplified with the following matrix notation

$$f_j\left(\sum_{j}^{n} w_{ij} - \sum_{i,j}^{n,n} w_{ij}\right) f_i = f^T(D-W)f \tag{4.5}$$

$$= f^T\mathcal{L}f \tag{4.6}$$

where $\mathcal{L}$, the graph Laplacian estimate of $\mathcal{M}$, is the regularization functional that punishes variations (via the L_2-norm) in local regions using a similarity measure W defined based on a neighborhood on $\mathcal{M}$ (von Luxburg (2007)). A common example is the Gaussian

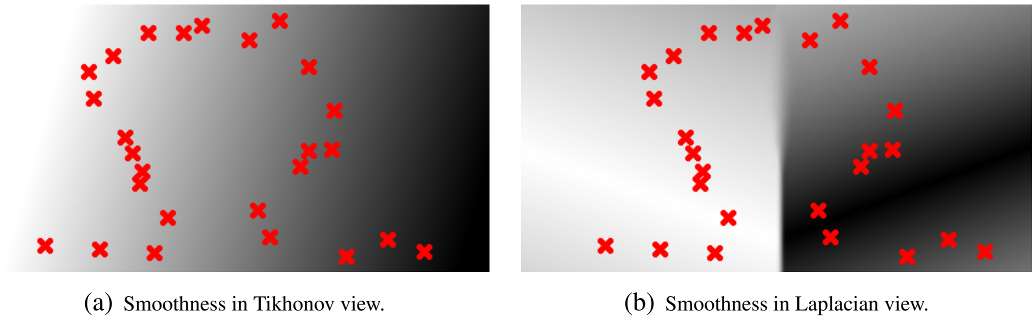
$$W_{ij} = w_{ij} := \exp\left(\frac{-\|x_i - x_j\|^2}{\sigma^2}\right), \tag{4.7}$$

where σ is often defined by the median distance between points.

When using the graph Laplacian in practice, it is generally normalized with $D^{1/2}$ on both sides

$$\tilde{\mathcal{L}} = D^{-1/2}(D-W)D^{-1/2}. \tag{4.8}$$

Use of $\tilde{\mathcal{L}}$ instead of $\mathcal{L}$ provides certain theoretical guarantees (von Luxburg et al. (2004)) and to better convergence properties of $\mathcal{L}$ to the Laplace-Beltrami operator (Belkin et al. (2006)). Thus from now on when we refer to $\mathcal{L}$, the $\tilde{\mathcal{L}}$ is implied.



(a) Smoothness in Tikhonov view.

(b) Smoothness in Laplacian view.

Figure 4.4: *Comparison of Tikhonov and Laplacian regularization, where in (a) the function changes slowly over the ambient space, as indicated by the slow change in gray values. If samples have very different gray values, they lie far apart. (b) shows a function that is unsmooth as measured by Tikhonov, but smooth measured by the Laplacian of the samples. Samples that lie really close together have similar gray values, but for samples that are not as close, the values can differ based on the lower-dimensional manifold estimate Laplacian is considering.*

In essence, we have defined a graph estimate $\mathcal{L}$ of the manifold $\mathcal{M}$ by connecting the data points (sampled from $\mathcal{M}$) based on their distance-weighted pairwise differences, in other words by the the distance-weighted edges that are encoded in $W_{i,j}$.

This results in a regularization term $\mathcal{L}$ with the same idea as that behind Tikhonov regularization discussed in Section 3.3.1, except instead of penalizing large variations in the projection function with respect to the ambient space, the Laplacian regularization term penalizes large variations (large gradients) in a local region only. This could result in a globally oscillating projection function, but locally smooth, because it is smooth with respect to the lower-dimensional manifold structure. Figure 4.4 illustrates how a function appears from the perspective of Laplacian regularization compared with Tikhonov regularization.

4.2 Semi-supervised Laplacian Regularization of KCCA

We can regularize KCCA using Laplacian regularization in a semi-supervised framework by calculating the graph Laplacian from the dataset containing both labeled and unlabeled samples, as shown above. Assume we have the same 2-modality data set, where we have additional data for both of the modalities but do not know the correspondences of these data. We have the paired data set of training data with correspondence $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ and the additional samples without correspondences, $\{x_{n+1}, \ldots, x_p\}$ and $\{y_{n+1}, \ldots, y_p\}$. With these, our two augmented datasets are the X modality $\dot{X} = \{x_1, \ldots, x_p\} \in \mathbb{R}^p$, and the Y modality $\dot{Y} = \{y_1, \ldots, y_n\} \in \mathbb{R}^n$. The kernel matrices of these two data sets are denoted by

$$K_{xx} = \Phi(X)^T\Phi(X) \tag{4.9}$$
$$K_{\dot{x}x} = \Phi(\dot{X})^T\Phi(X) \tag{4.10}$$
$$K_{\dot{x}\dot{x}} = \Phi(\dot{X})^T\Phi(\dot{X}) \tag{4.11}$$

and

$$K_{yy} = \Phi(Y)^T\Phi(Y) \tag{4.12}$$
$$K_{\dot{y}y} = \Phi(\dot{Y})^T\Phi(Y) \tag{4.13}$$
$$K_{\dot{y}\dot{y}} = \Phi(\dot{Y})^T\Phi(\dot{Y}). \tag{4.14}$$

We calculate the corresponding graph Laplacians $\mathcal{L}_{\dot{x}}$ and $\mathcal{L}_{\dot{y}}$, to the datasets $\dot{X}$ and $\dot{Y}$, respectively. These terms can be introduced for Laplacian regularization of the two data domains into the optimization problem for Tikhonov regularized KCCA shown in Section 3.3.1, Equation (3.67):

$$\max_{\alpha,\beta} \frac{\alpha^T K_{\dot{x}x} K_{y\dot{y}} \beta}{\sqrt{\alpha^T \left(K_{\dot{x}x} K_{x\dot{x}} + R_{\dot{x}}\right) \alpha \beta^T \left(K_{\dot{y}y} K_{y\dot{y}} + R_{\dot{y}}\right) \beta}}, \tag{4.15}$$

where the Tikhonov and Laplacian regularizers for both modalities are contained in $R_{\dot{x}}$ and $R_{\dot{y}}$

$$R_{\dot{x}} = \varepsilon_x K_{\dot{x}\dot{x}} + \frac{\gamma_x}{p_x^2} K_{\dot{x}\dot{x}} \mathcal{L}_{\dot{x}} K_{\dot{x}\dot{x}} \tag{4.16}$$
$$R_{\dot{y}} = \varepsilon_y K_{\dot{y}\dot{y}} + \frac{\gamma_y}{p_y^2} K_{\dot{y}\dot{y}} \mathcal{L}_{\dot{y}} K_{\dot{y}\dot{y}} \tag{4.17}$$

where the ε_x and ε_y are the Tikhonov regularization parameters, and $\frac{\gamma_x}{p_x^2}$ and $\frac{\gamma_x}{p_x^2}$ are the normalized Laplacian regularization parameters, all of which control the degree of penalization (regularization) of the respective L_2-norms in each Hilbert space ($\mathcal{H}_x$ and $\mathcal{H}_y$).

Following identical steps as enumerated with Tikhonov regularized KCCA in Section 3.3.1, we can write the above regularized expression as a constrained optimization problem, where both parts of the denominator are set to 1, and we maximize

$$\alpha^T K_{\dot{x}x} K_{y\dot{y}} \beta \tag{4.18}$$

subject to:

$$\alpha^T K_{\dot{x}x} K_{x\dot{x}} \alpha + \varepsilon_x \alpha^T K_x \alpha + \frac{\gamma_x}{p_x^2} \alpha^T K_{\dot{x}\dot{x}} \mathcal{L}_{\dot{x}} K_{\dot{x}\dot{x}} \alpha = 1 \tag{4.19}$$

$$\beta^T K_{\dot{y}y} K_{y\dot{y}} \beta + \varepsilon_y \beta^T K_y \beta + \frac{\gamma_y}{p_y^2} \beta^T K_{\dot{y}\dot{y}} \mathcal{L}_{\dot{y}} K_{\dot{y}\dot{y}} \beta = 1, \tag{4.20}$$

which, for simplicity in writing, can be equivalently expressed as the maximization of

$$\alpha^T K_{\dot{x}x} K_{y\dot{y}} \beta \tag{4.21}$$

subject to:

$$\alpha^T K_{\dot{x}x} K_{x\dot{x}} \alpha + \alpha^T R_{\dot{x}} \alpha = 1 \tag{4.22}$$

$$\beta^T K_{\dot{y}y} K_{y\dot{y}} \beta + \beta^T R_{\dot{y}} \beta = 1, \tag{4.23}$$

with regularizers

$$R_{\dot{x}} = \varepsilon_x K_{\dot{x}\dot{x}} + \frac{\gamma_x}{p_x^2} K_{\dot{x}\dot{x}} \mathcal{L}_{\dot{x}} K_{\dot{x}\dot{x}} \tag{4.24}$$

$$R_{\dot{y}} = \varepsilon_y K_{\dot{y}\dot{y}} + \frac{\gamma_y}{p_y^2} K_{\dot{y}\dot{y}} \mathcal{L}_{\dot{y}} K_{\dot{y}\dot{y}}. \tag{4.25}$$

This constrained optimization problem has the corresponding Lagrangian function:

$$L(\lambda_\alpha, \lambda_\beta, \alpha, \beta) = \alpha^T K_{\dot{x}x} K_{y\dot{y}} \beta - \lambda_\alpha (\alpha^T K_{\dot{x}x} K_{x\dot{x}} \alpha + \alpha^T R_{\dot{x}} \alpha - 1) - \lambda_\beta (\beta^T K_{\dot{y}y} K_{y\dot{y}} \beta + \beta^T R_{\dot{y}} \beta - 1). \tag{4.26}$$

The minimization of this Lagrangian function is obtained as follows

$$\frac{\partial L}{\partial \lambda_\alpha}(\lambda_\alpha, \lambda_\beta, \alpha, \beta) = 0 \tag{4.27}$$

$$\Leftrightarrow K_{\dot{x}x} K_{y\dot{y}} \beta = 2\lambda_\alpha (K_{\dot{x}x} K_{x\dot{x}} \alpha + \alpha^T R_{\dot{x}} \alpha) \tag{4.28}$$

$$\frac{\partial L}{\partial \lambda_\beta}(\lambda_\alpha, \lambda_\beta, \alpha, \beta) = 0 \tag{4.29}$$

$$\Leftrightarrow K_{\dot{y}y}K_{x\dot{x}}\alpha = 2\lambda_\beta(K_{\dot{y}y}K_{y\dot{y}}\beta + \beta^T R_{\dot{y}}\beta), \tag{4.30}$$

from which (following the same algebraic steps in 3.3) we can derive the following generalized eigenvalue problem:

$$\begin{pmatrix} K_{\dot{x}x}K_{x\dot{x}} + R_{\dot{x}} & K_{\dot{x}x}K_{y\dot{y}} \\ K_{\dot{y}y}K_{x\dot{x}} & K_{\dot{y}y}K_{y\dot{y}} + R_{\dot{y}} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \lambda \begin{pmatrix} K_{\dot{x}x}K_{x\dot{x}} + R_{\dot{x}} & 0 \\ 0 & K_{\dot{y}y}K_{y\dot{y}} + R_{\dot{y}} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}. \tag{4.31}$$

As with Tikhonov regularized KCCA outlined in Chapter 3, the solution to the above eigenvalue problem yields the coefficient vectors α^1 and β^1, which maximize the correlation between the Φ_X and Φ_Y data projections. Interesting now is that with the addition of Laplacian regularization, these projections of maximal correlation will also be smooth along the manifold (Blaschko et al. (2008)). We acquire a numerical solution for the first weight vectors $\tilde{u}_x^1$ and $\tilde{u}_y^1$ by Equations (3.41) and (3.42). These vectors in the Hilbert spaces are analogous to the u_x^1 and u_y^1 in the input spaces resulting from Equation (2.57). As with the linear dimensionality reduction methods, to obtain the further directions of maximal correlation, we simply take the next eigenvectors α^i and β^i corresponding to the next highest eigenvalues λ^i, leading to weight vector solutions of $\tilde{u}_x^i$ and $\tilde{u}_y^i$ that are smooth with respect to the ambient space (Tikhonov regularization) and with respect to the [graph Laplacian estimated] manifold structure (Laplacian regularization).

4.2.1 Properties of Semi-supervised KCCA

The two-modality case of semi-supervised KCCA can be generalized to incorporate multiple modalities as shown for standard KCCA in Chapter 2, except where each has additional data without known correspondence. See Blaschko et al. (2008).

Chapter 5

Methods and Materials

5.1 Functional Magnetic Resonance Imaging (fMRI) Data

fMRI studies entail having a human volunteer lie in an fMRI scanner (see Figure 5.1) and showing him or her some form of stimulus, and recording their brain activation patterns in the form of 3D images every few seconds (see Figure 5.2). The goal of these studies is to localize the main regions of brain activity corresponding to a given stimulus. fMRI data is well tuned to subspace methods of dimensionality reduction given that data are already well-aligned due to the nature of the acquisition process. Thus the problem is inherently one of dimensionality reduction – reduction of the pixels in a high-dimensional brain image to a reasonably small region of localized pixels, corresponding to a given aspect of the input stimulus (see Figure 5.3). Particularly when the goal is to localize brain activity during natural visual processing tasks, such as watching a video, the analysis of this data is a very challenging task for machine learning.

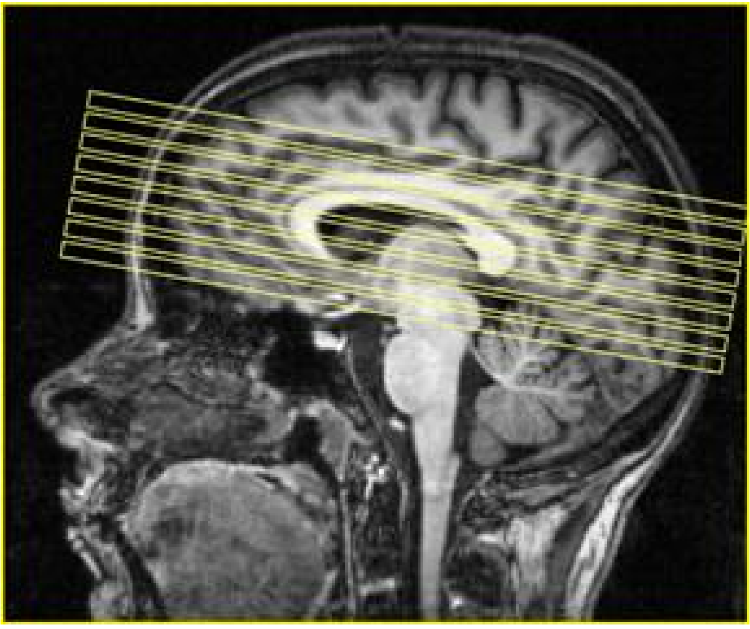
The studies aiming to localize brain activity, or reduce the activity to the main dimensions of activity (brain regions consisting of 3-dimensional pixels, voxels), face a number of computational and practical problems. First, fMRI studies yield extremely high-dimensional data, i.e. 36,000 dimensions per image (at a given time-point), so to notice any patterns in the data, the dimensions need to be drastically reduced. Second, the safe amount of time that a human may remain in an fMRI scanner due to the strength of the magnet is limited, and the demand of such fMRI scanning facilities is high. This leads to the number of time points recordable at a given time period to be very limited, and with an n-size much smaller than that of the dimensions of the data, the learning problem is very susceptible to overfitting. Finally, when using natural visual stimuli, one needs labels for every few frames of the movie, indicating the content of said frames' stimulus during

(a) Functional Magnetic Resonance Imaging (fMRI) Scanner.

Figure 5.1: *A human volunteer lies in an fMRI scanner as they are shown some sort of stimulus and his/her brain activity is recorded in terms of the blood oxygen level-dependent (BOLD) response.*

the corresponding fMRI data acquisition. As the natural stimuli are complex, these labels need to be subjectively generated by several human observers. This process requires at least five observers to watch each stimulus (i.e. a movie) and score the content of the frames every few seconds, encoding the degree to which the stimulus aspects of interest are present in the frame (i.e. faces, bodies). This is a very time consuming and expensive process, and data without labels is relatively less expensive.

Thus, the current work seeks to address the above problems in the analysis of fMRI data as well as to improve its inferential performance, by performing dimensionality reduction via KCCA in a semi-supervised (Laplacian regularized) framework. The approach of KCCA with fMRI data has only recently been explored before by Hardoon et al. (2007), but never in a semi-supervised learning framework. We hypothesize that this framework will not only allow maximum use of all available data, even those without known correspondences (stimulus labels), and thereby aid the problem of overfitting, but also improve inference based on use of manifold regularization.



(a) Sample fMRI brain image.

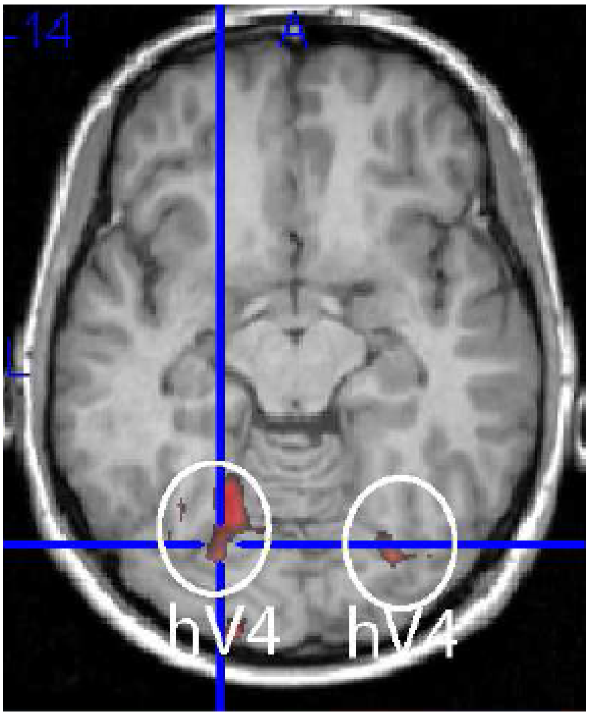
Figure 5.2: *This figure shows how a sample fMRI brain image appears, and how the brain is recorded in subdivided "slices" during acquisition. These subdivided slices are recorded through the entire brain at every given time point, then concatenated to form one brain image vector.*

5.1.1 Blood Oxygen Level Dependent (BOLD) Response

fMRI data measures the hemodynamic response (change in blood flow) related to neural activity in the brain, or the BOLD response. The changes in the BOLD signal has been implicated in many studies, identifying its coupling with blood flow and metabolic rate, and indicating its correlation with neural activity (Ulmer and Jansen (2010)). BOLD effects are measured by acquisition of contrast-weighted volumetric images, wherein each 3-dimensional voxel we obtain represents a 3 millimeter cube of brain tissue BOLD response (Ulmer and Jansen (2010)).

5.1.2 Data Acquisition

All of the fMRI data used was acquired at the Max Planck Institute for Biological Cybernetics, using a Siemens 3 Tesla (3T) TIM scanner. All data sets obtained consisted of 350 time-slices of 3-dimensional fMRI brain volumes (voxels), separated by 3.2 seconds, temporal resolution (TR). From this set of 350 time-slices, the first and last three had to be removed due to the recording artifacts they introduced.



(a) Localized regions of brain activity after successful dimensionality reduction.

Figure 5.3: *In this figure we see the localized regions of brain activity during a given stimulus, the reduced dimensions (pixels/voxels) of activity. This is a visualization of a given ”slice” through the brain, as illustrated in Figure 5.2.*

5.1.3 Natural Viewing Data

Recently there has been a great surge in interest in assessing natural visual processing. Specifically, in the brain activity occurring during a natural and complex setting, such as having the human volunteer watch a video in the scanner in order to gain insight into the brain processes and connectivity underlying more natural processing. (i.e. see Figure 5.1). Previously brain activity was analyzed in a more controlled setting, showing unnatural and limited stimuli, which leads to easier data analysis. As such, analysis of this data has faced a number of problems, as outlined earlier in this Chapter.

The problem of analyzing natural fMRI viewing data has been approached by neuroscientists from various routes: linear regression was used to identify brain areas that correlate with particular labels in the movie (Bartels and Zeki (2004b)), the perceived content was inferred based on brain activity (Hasson et al. (2004)), data-driven methods were used to subdivide the brain into units with distinct response profiles (Bartels and Zeki (2004a)), and correlation across subjects was used to infer stimulus-driven brain processes at different timescales (Hasson et al. (2008)).

As such, we have a ground-truth with which we can compare (1) what regions our method of SSKCCA learns and localizes, and (2) how well it learns, what the empirical performance is in terms of how we evaluate the method (see Section 6.1). As described above, previous neuroscientific studies have implicated the regions we can expect the method to learn should our hypotheses about the method be fulfilled.

5.1.4 Resting State Activity Data

In order to explore a another semi-supervised setting in which unlabeled data are acquired as a biproduct of other fMRI studies with the labeled data, resting state data was explored as a potential additional source of unlabeled data. The inclusion of such data would allow researchers to maximize the use of all acquired data, without increasing costs of a) data acquisition and/or b) costly data labels.

Resting state activity has drawn the attention of neuroscientists for more than a decade (Biswal et al. (1995)). This type of fMRI data is defined as brain activation which occurs in the absence of any task or stimuli, and is generally measured in awake subjects during prolonged fMRI scanning sessions, as described above. The instruction given to the volunteers is simply to close their eyes and to do nothing. The basic idea is that spontaneous fluctuations of neural activity in the brain may reveal some fundamental characteristics of brain function. These aspects could be functional as well as structural.

5.2 Stimuli Labels

The continuous label time-series were obtained using two separate methods: via the frame-by-frame computer analysis of the movie (Bartels et al. (2007)), and through extensive subjective ratings averaged across an independent set of five human observers (Bartels and Zeki (2004a)). The computer-derived labels indicated luminance change over time (temporal contrast) and visual motion energy (i.e. the fraction of temporal contrast that can be explained by motion in the movie). The former were inexpensive to obtain whereas the latter were considerably costly.

The human-derived labels were the time-extensive and tedious ones to obtain. They were acquired from five human observers, each of whom had to watch the movie five times to provide a rating every 3.2 seconds for the five respective labels. The five labels consisted of: (1) the intensity of subjectively experienced color, the degree to which (2) faces, (3) language, (4) motion, and (5) human bodies were present in the movie. In prior studies, each of these labels had been shown to correlate with brain activity in particular and distinct sets of areas specialized to process the particular label in question (Bartels and Zeki (2004a); Bartels et al. (2007)).

5.3 Pre-processing Methods

The imaging data were pre-processed using standard procedures using the Statistical Parametric Mapping (SPM) toolbox before analysis (Friston et al. (2007)). Included was a slice-time correction to compensate for acquisition delays between slices, a spatial realignment to correct for small head-movements, a spatial normalization to the SPM standard brain space (near MNI).

The data used in the first set of experiments described in Section 6.3 was spatially smoothed using a Gaussian filter of 6 mm full width at half maximum (FWHM), whereas for the goal of comparing across subjects in the experiments described in Section 6.4, spatial smoothing was performed using a Gaussian filter of 12 mm full width at half maximum (FWHM). Subsequently, all images were skull-and-eye stripped and the mean of each time-slice was set to the same value (global scaling). A temporal high-pass filter with a cut-off of 512 s was applied, as well as a low-pass filter with the temporal properties of the hemodynamic response function (HFR), in order to reduce temporal acquisition noise.

Additionally, to correct for the delay of the peak of the BOLD response to a stimulus mentioned earlier, the stimulus time-series (the labels corresponding to one of the video stimuli) was convolved with a temporal filter modeling the dynamics of a generic hemodynamic response function (HRF), the same as with the fMRI data (Friston et al. (2007)).

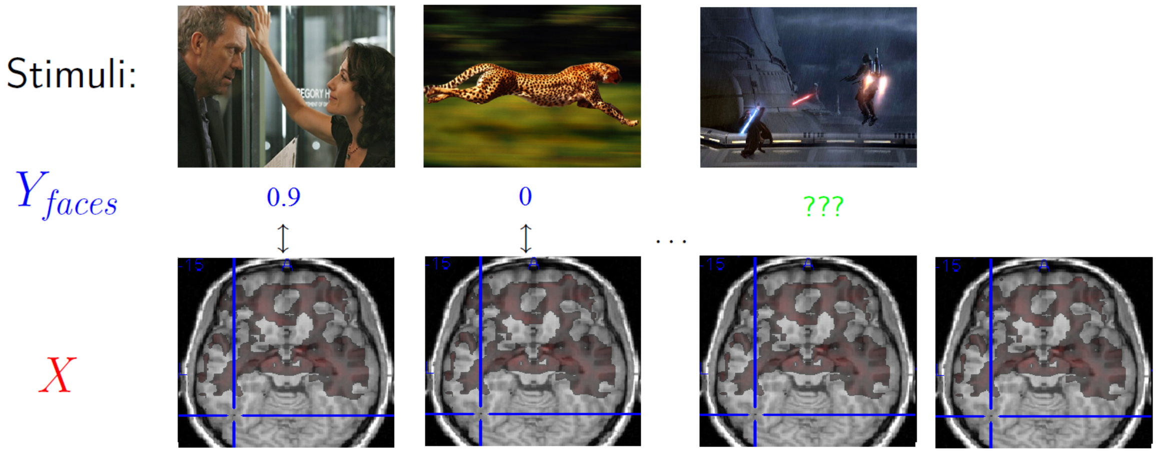
5.4 Paired Data and Notation

We will now formalize the above setting as necessary for the two-modality case of KCCA and for SSKCCA. The interesting part about the above described setting is that the data are available in two modalities (fMRI data and stimulus labels), making this a perfect candidate for KCCA.

The $\mathcal{X}$ modality is composed of the fMRI data. The active viewing fMRI data (Sections 5.1.3) consisted of $n = 344$ time-slices of brain activation images acquired during the viewing of *two movies* of 18.5 minutes each in length. The data from the first movie, where the correspondences with the $\mathcal{Y}$ modality are known, is denoted with $X = (x_1, \ldots, x_n) \in \mathbb{R}^{n \times 36,268}$. The data acquired during the second movie, the additional data source of unlabeled active viewing fMRI data, is denoted $\tilde{X} = (x_{n+1}, \ldots, x_p) \in \mathbb{R}^{n-p \times 36,268}$. The additional source of "unlabeled" data, the resting state data (Section 5.1.4) is denoted $\hat{X} = (x_{n+1}, \ldots, x_p) \in \mathbb{R}^{p-n \times 36,268}$.

The $\mathcal{Y}$ modality consisted of the corresponding labels (Section 5.2) of the first movie's content (corresponding to activity in X). These stimulus labels were obtained from the scores of five human observers for *one movie*, $Y\{y_1, \ldots, y_n\} \in [0, 1]$, where $n = 344$. Labels were not obtained for the second movie given the goals of this exploration.

An illustration of this setting is provided in Figure 5.4.



(a) Illustration of the setting of the various types of data.

Figure 5.4: *Examples of some stimuli frames shown during fMRI acquisition are shown on the top of the figure, where below that are the corresponding scores for one of the labels, faces, to those frames if known. Below that are the brain activation patterns corresponding to the viewing of the above frames. The two sets of images (stimuli and fMRI data) on the left are examples of paired data, where* $D = \{(x_1, y_1 = .9), (x_2, y_2 = 0)\}$*. This is the expensive labeled data. The three remaining images on the right are illustrations of (1) unlabeled fMRI data, where the correspondence of the face label content of the frame is unknown, and (2) resting state fMRI data, where activity is recorded in the absence of a stimulus.*

Chapter 6

Experiments

The work presented in this chapter has yielded a few publications, namely that presented partially in Shelton et al. (2009a,b); Blaschko et al. (submitted, 2009).

All experiments consisted of the most general model of SSKCCA in Expression (4.15) with appropriate model-selected model parameters to test different performance aspects of the algorithm with our data sets. As we have additional data for only one of the two modalities, specifically we have p samples in $\dot{X}$ for the $\mathcal{X}$ modality, but only n ($n < p$) in Y for the $\mathcal{Y}$ modality, where the second modality only consists of stimuli labels corresponding to $\mathcal{X}$. With these two datasets, SSKCCA in Expression (4.15) is reduced to

$$\max_{\alpha,\beta} \frac{\alpha^T K_{\dot{x}x} K_{yy} \beta}{\sqrt{\alpha^T \left(K_{\dot{x}x} K_{x\dot{x}} + R_{\dot{x}}\right) \alpha \beta^T \left(K_y^2\right) \beta}}, \tag{6.1}$$

with regularizers

$$R_{\hat{x}} = \varepsilon_x K_{\dot{x}\dot{x}} + \frac{\gamma_x}{m_x^2} K_{\dot{x}\dot{x}} \mathcal{L}_{\dot{x}} K_{\dot{x}\dot{x}}. \tag{6.2}$$

Note that no regularizers on Y are needed due to the nature of the Y modality. Smoothness in the ambient space and with respect to a manifold does not make sense when the data consist only of continuous labels.

6.1 Evaluation Methodology

In order to evaluate the performance of KCCA on the fMRI data described in Section 5.4, and the effect of semi-supervised Laplacian regularization on the performance of KCCA, I have evaluated three variants of the algorithm:

- In the first variant, we have run Tikhonov regularized KCCA without any Laplacian regularization, by setting $\gamma_x = 0$ in Expression (6.1), and using only the paired datasets X and Y.
- The second variant consists of Laplacian regularization where the empirical Laplacian matrix (graph Laplacian $\mathcal{L}$) was computed using only data for which correspondences between the $\mathcal{X}$ and $\mathcal{Y}$ modalities were known, using only the X set.
- In the final variant, we used full semi-supervised Laplacian regularization, where the graph Laplacian estimate of the manifold was calculated using all available training data, $\dot{X}$.

These methods will be briefly revisted and outlined in the description of the two experimental sets.

We evaluate the performance of the algorithms quantitatively based on the magnitude of the correlation of the learned projections of $\tilde{u}_x^1$ and $\tilde{u}_y^1$, across all of the experimental manipulations (different model parameter settings and data sets) and across all stimuli labels. To evaluate the performance of these experiments on unseen data and obtain testing performance values, we have run five-fold cross validation. This entails dividing the data into five set of equal n's and holding one of them out, the "hold-out" set. Then we train the model our on four sets, and test on the remaining hold-out set, obtaining one hold-out correlation between the learned projections of $\tilde{u}_x^1$ and $\tilde{u}_y^1$. This is repeated on all permutations of the data divisions until we have obtained five hold-out correlations of the data projections.

In all cases, we have used linear kernels on both the input and output spaces, X and Y. This is such that we can interpret the output $\tilde{u}_x^1$ of the learned projection function, f_x (refer to Chapter 4 for details), by reconstructing the vector as a learned map of the brain regions implicated in the various visual processing tasks.

The graph Laplacian's Laplacian matrix was computed using a Gaussian neighborhood kernel ($W_{ij} = w_{ij} := \exp\left(\frac{-\|x_i - x_j\|^2}{\sigma^2}\right)$) with the bandwidth parameter σ set to the median distance between all pairs of training data with and without correspondences ($\sigma = median_{i,j}\|x_i - x_j\|$). We have used the symmetric normalized Laplacian $\mathcal{L} = D^{-\frac{1}{2}}(D - W)D^{\frac{1}{2}}$, where D is the diagonal matrix whose entries are the row sums of the similarity matrix, W (see Section 4.1.1).

6.2 Model Selection

Regularization parameters need to be carefully selected in order to tune the model to the data at hand, and to trade-off between testing error and training error. This is particularly important in the case of Laplacian regularization as explained in Chapter 4. We have used two model selection criteria to optimize over the regularization parameters of ε_x and γ_x, for Tikhonov and Laplacian regularization, respectively. Both criteria are used as the inner loop of a grid search.

The first method of model selection used was cross validation. We selected the model parameters that maximize a five-fold cross validation estimate of the empirical correlation (using only the training data).

Although cross validation is a thorough method of model selection, it is both computationally and statistically inefficient. Thus we have also evaluated a model selection criterion proposed by Hardoon et al. (2004) which is intended to replicate the results of cross validation with a computationally more efficient approach. This method consists of creating a random permutation of the data correspondences and running the SSKCCA generalized eigen value problem with both the data sets, unpermuted data and with the permuted data. The parameter setting taken to be the optimum with is that with the maximum norm of the difference of the spectra of the two SSKCCA eigenvalue problems (that computed with the permuted data compared with that computed with the unpermuted data).

6.3 Active Viewing Experiments

The first set of experiments had a number of goals. First they aimed to ascertain whether KCCA is indeed a suitable algorithm for dimensionality reduction of fMRI data. Furthermore, the main goal was to determine (a) whether the manifold assumption holds with this particular type of high-dimensional data, and (b) if Laplacian regularization improves performance of KCCA in terms of hold-out correlations and/or in terms of neuroscientific interpretation of the learned projection coefficients. Our hypotheses predict that the manifold assumption will hold with fMRI data, and that Laplacian regularization will improve inference performance of KCCA.

As such, these experiments used *only* the two sets of active viewing data from one human volunteer and stimuli labels described in 5.4. The KCCA experiment variants consisted of:

(1) KCCA with Tikhonov regularization → labeled data only (supervised),

(2) KCCA with Tikhonov and Laplacian regularization → labeled data only (supervised),

(3) SSKCCA with Tikhonov and Laplacian regularization → labeled and unlabeled data (semi-supervised).

6.3.1 Results

The quantitative and qualitative results are presented in the first two sections below, and these results are summarized and discussed in the section to follow.

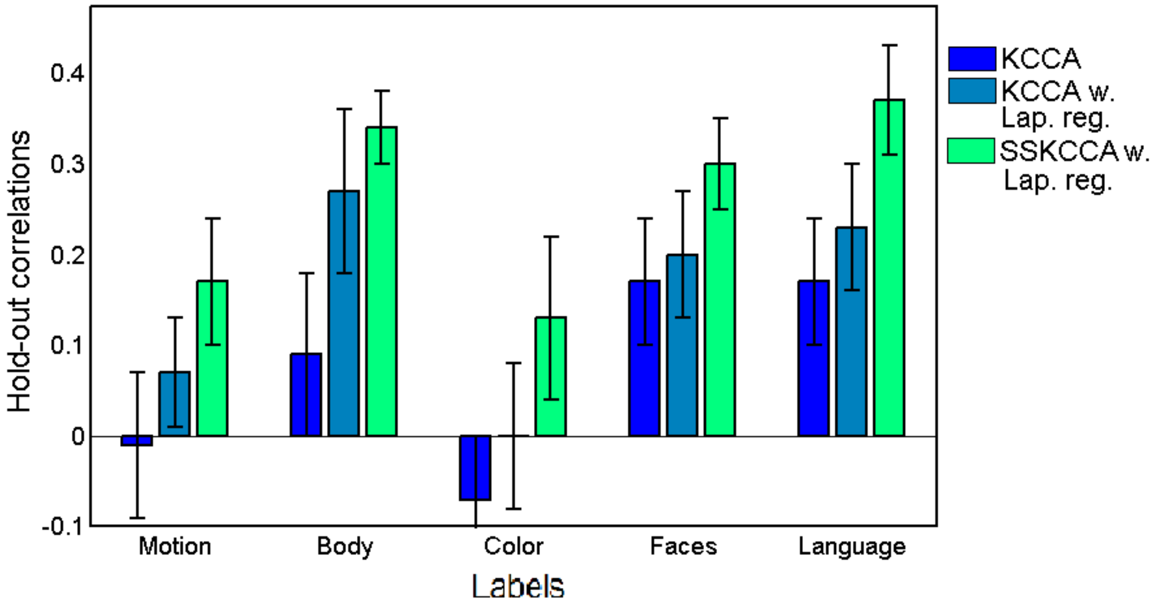
Quantitative Results

The visual content of the stimulus is quantified in six variables: Motion, Temporal Contrast, Human Body, Color, Faces, and Language. We have repeatedly run all three variants of the experimental setup (Section 6.1) setting our output space to each individual variable. The results for the cross validation model selection are shown in Table 6.1, and the results for the spectral model selection are shown in Table 6.2. We have additionally run experiments with multi-variate output by grouping several of the variables into three groups: {Visual motion energy, Body, Color}; {Motion, Faces}; and {Motion, Visual motion energy, Color, Faces}. The results of these experiments using the spectral model selection are shown in Table 6.3.

Table 6.1: *Mean holdout correlations across the six variables in all experiments with five-fold cross-validation. Experiment 1: KCCA using only data for which correspondences are known, X, and Tikhonov regularization. Experiment 2: Laplacian regularization where the graph Laplacian $\mathcal{L}$ is estimated using only data for which correspondences are known, X. Experiment 3: full semi-supervised Laplacian regularization, $\mathcal{L}$ calculated using $\dot{X}$. Semi-supervised Laplacian regluarized KCCA yields the best performance in all cases.*

	Motion	Temporal Contrast	Human Body	Color	Faces	Language
Exp 1	-0.012 ± 0.081	0.042 ± 0.065	0.095 ± 0.086	-0.075 ± 0.069	0.173 ± 0.073	0.172 ± 0.070
Exp 2	0.065 ± 0.066	0.088 ± 0.084	0.274 ± 0.093	-0.002 ± 0.079	0.203 ± 0.075	0.231 ± 0.074
Exp 3	0.170 ± 0.074	0.116 ± 0.101	0.340 ± 0.043	0.128 ± 0.089	0.303 ± 0.054	0.365 ± 0.057

The results of Table 6.2, the mean hold-out correlations for each experiment and each of the man-made labels (Section 5.2) are presented in Figure 6.1.



(a) Mean hold-out correlations across all three experiments for all man-made stimulus labels.

Figure 6.1: *Experiment 1: KCCA using only data for which correspondences are known, X, and Tikhonov regularization. Experiment 2: Laplacian regularization where the graph Laplacian $\mathcal{L}$ is estimated using only data for which correspondences are known, X. Experiment 3: full semi-supervised Laplacian regularization, $\mathcal{L}$ calculated using $\dot{X}$. Semi-supervised Laplacian regluarized KCCA yields the best performance in all cases.*

Table 6.2: *Mean holdout correlations across the six variables in all experiments with the spectral model selection criterion of Hardoon et al. (2004). Experiment 1: KCCA using only data for which correspondences are known, X, and Tikhonov regularization. Experiment 2: Laplacian regularization where the graph Laplacian $\mathcal{L}$ is estimated using only data for which correspondences are known, X. Experiment 3: full semi-supervised Laplacian regularization, $\mathcal{L}$ calculated using $\dot{X}$. Semi-supervised Laplacian regluarized KCCA yields the best performance in all cases.*

	Motion	Temporal Contrast	Human Body	Color	Faces	Language
Exp 1	-0.012 ± 0.081	0.042 ± 0.065	0.095 ± 0.086	-0.075 ± 0.069	0.173 ± 0.073	0.172 ± 0.070
Exp 2	0.065 ± 0.066	0.088 ± 0.084	0.274 ± 0.093	-0.002 ± 0.079	0.203 ± 0.075	0.231 ± 0.074
Exp 3	0.170 ± 0.074	0.116 ± 0.101	0.340 ± 0.043	0.128 ± 0.089	0.303 ± 0.054	0.365 ± 0.057

Table 6.3: *Mean holdout correlations across the 3 multi-variate sets in all experiments with the spectral model selection criterion of Hardoon et al. (2004). Experiment 1: KCCA using only data for which correspondences are known, X, and Tikhonov regularization. Experiment 2: Laplacian regularization where the graph Laplacian $\mathcal{L}$ is estimated using only data for which correspondences are known, X. Experiment 3: full semi-supervised Laplacian regularization, $\mathcal{L}$ calculated using $\dot{X}$. Semi-supervised Laplacian regluarized KCCA yields the best performance in all cases.*

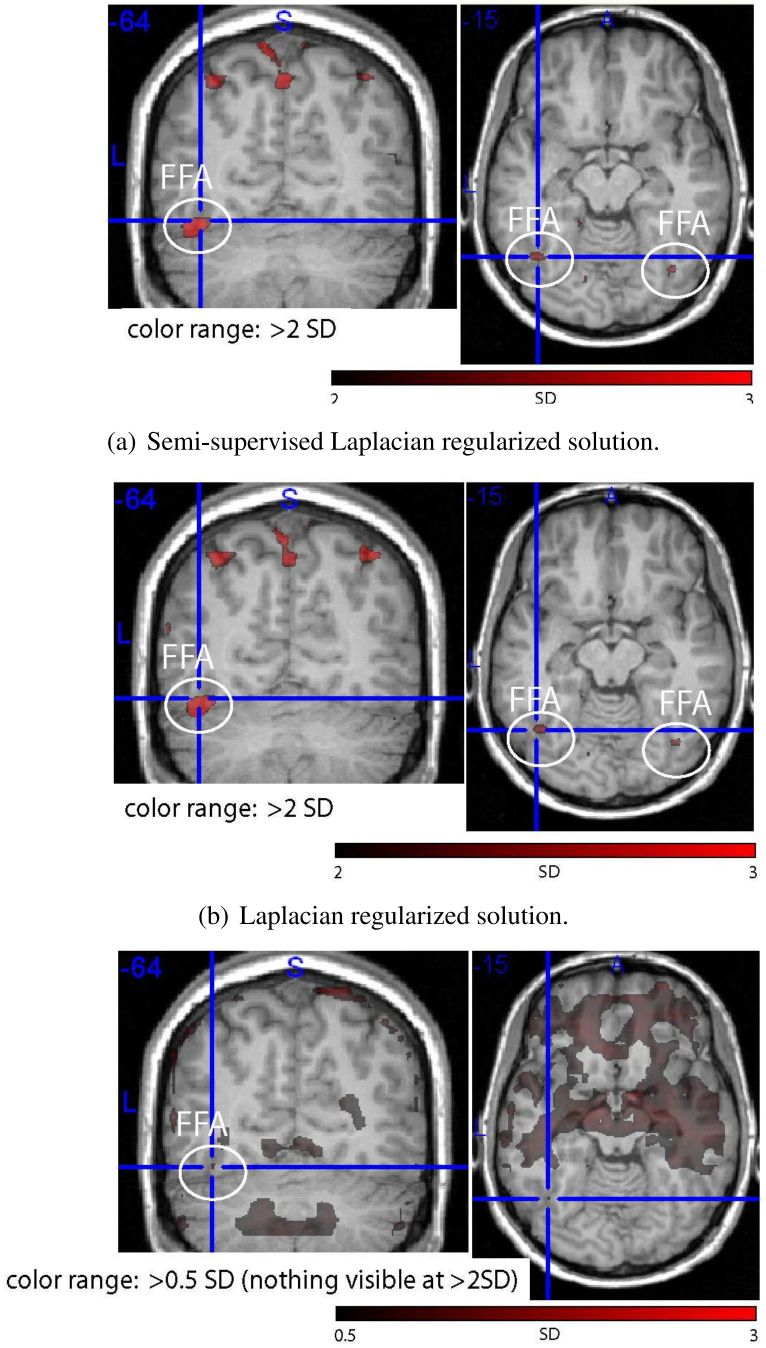
	Visual motion energy, Body, Color	Motion, Faces	Motion, Vis. mot. energy, Color, Faces
Experiment 1	0.1596 ± 0.0807	-0.0827 ± 0.0460	0.1167 ± 0.0785
Experiment 2	0.1873 ± 0.0879	0.0602 ± 0.0908	0.1498 ± 0.0827
Experiment 3	0.2844 ± 0.0716	0.1898 ± 0.0636	0.2528 ± 0.0579

Qualitative Results

As we have used linear kernels in all cases, we can interpret the outputs of the various KCCA experiments by visualizing the learned weights in the $\tilde{u}_x^1$. These are the coefficients assigned to different spatially localized brain regions in a given KCCA experiment. We show results for visual stimulus consisting of *Faces* in Figure 6.2, *Human body* in Figure 6.11, *Color* in Figure 6.4, and *Motion* in Figure 6.10. In Figure 6.6 we show results from multivariate output consisting of *Motion* and *Faces*.

Evaluation

Our hypotheses that this fMRI data set is a good candidate for KCCA and furthermore for Laplacian regularized semi-supervised KCCA were confirmed. The quantitative results displayed in the Tables and in Figure 6.1 depict prominent trends. In each variate condition, Laplacian regularization improved hold-out correlations above the fully-supervised variant, and the semi-supervised variants of KCCA yielded the highest hold-out correlations. These results suggest that Laplacian regularized KCCA can generalize better to new data than Tikhonov regularized KCCA, and that the manifold assumption holds with

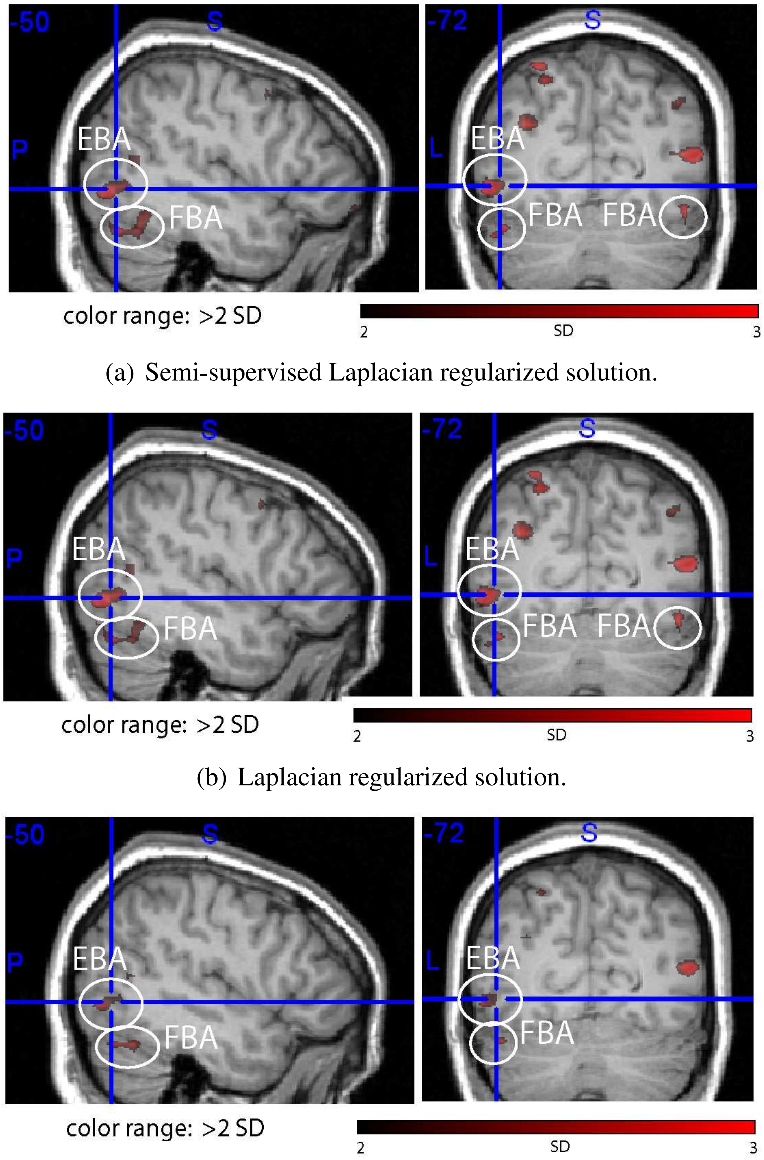


(a) Semi-supervised Laplacian regularized solution.

(b) Laplacian regularized solution.

(c) KCCA without Laplacian regularization.

Figure 6.2: *Faces: activation in the cortical region responsive to the visual perception of faces, the fusiform face area (FFA). Weight vectors are plotted over an anatomical image of the volunteers brain. Note that the semi-supervised Laplacian regularization led to the most specific and most significant weights in FFA.*

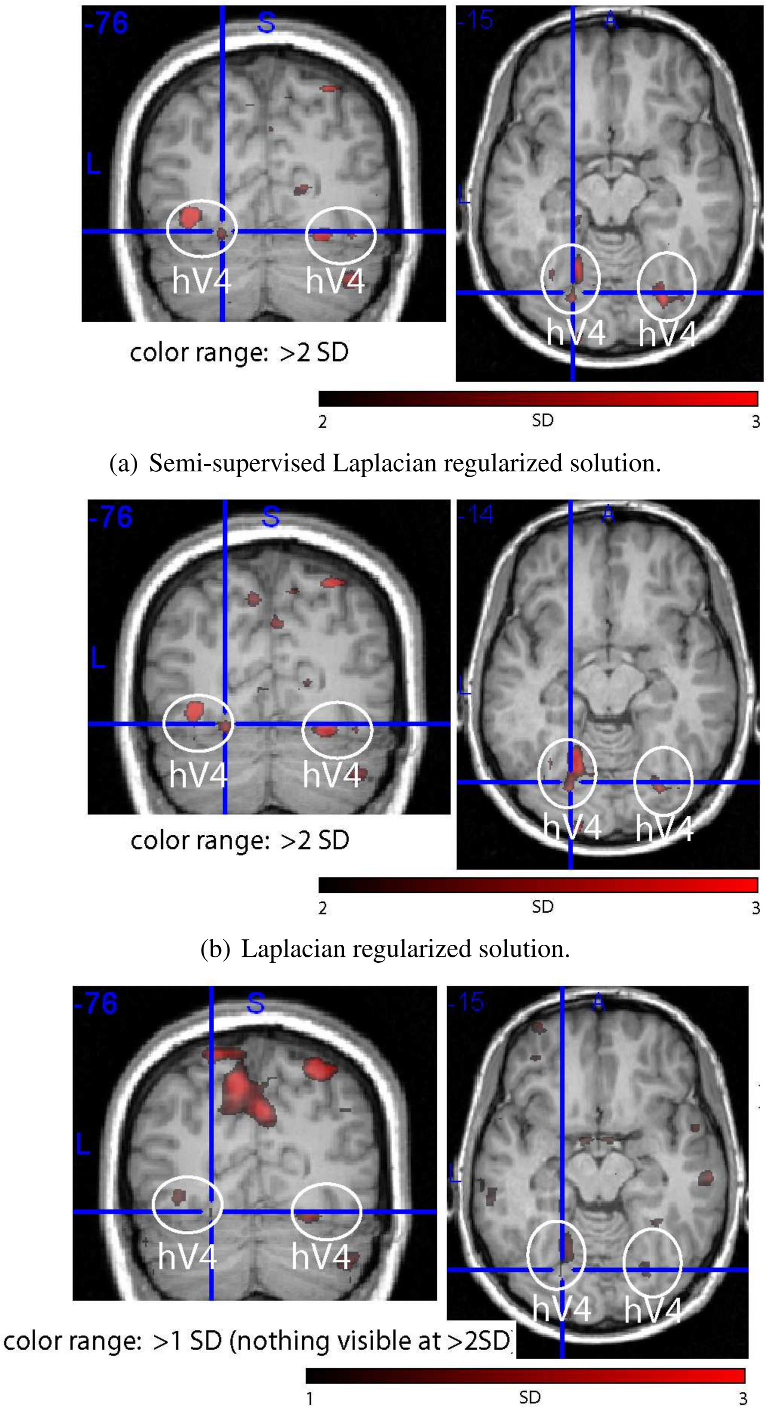


(a) Semi-supervised Laplacian regularized solution.

(b) Laplacian regularized solution.

(c) KCCA without Laplacian regularization.

Figure 6.3: *Human Body: activation in the cortical region responsive to the visual perception of human bodies, in the extrastriate body area (EBA) and in the fusiform body area (FBA). Same observation as in Figure 6.2.*

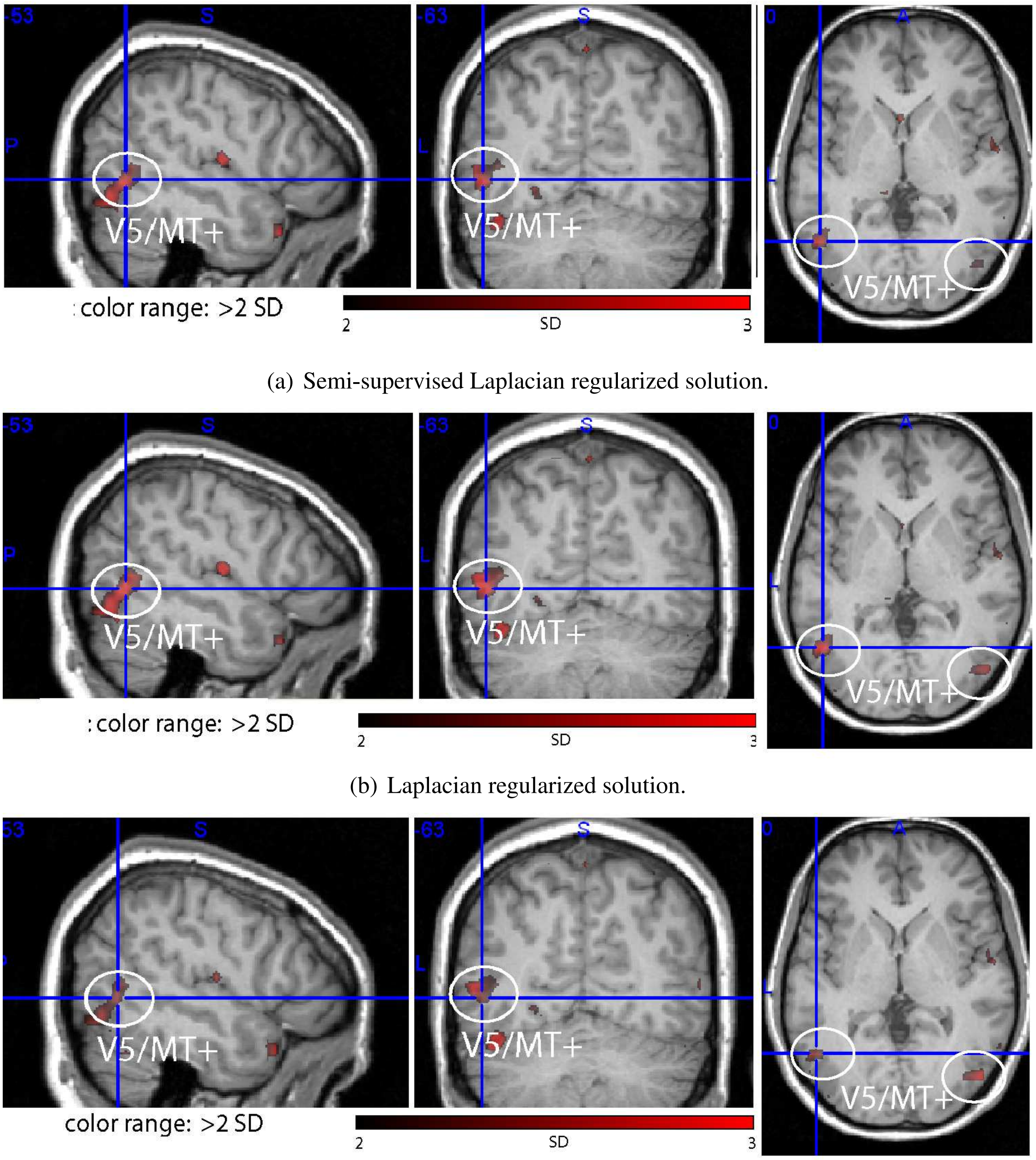


(a) Semi-supervised Laplacian regularized solution.

(b) Laplacian regularized solution.

(c) KCCA without Laplacian regularization.

Figure 6.4: *Color: activation in the color responsive cortex (human visual area 4, hV4). Same observation as in Figure 6.2.*

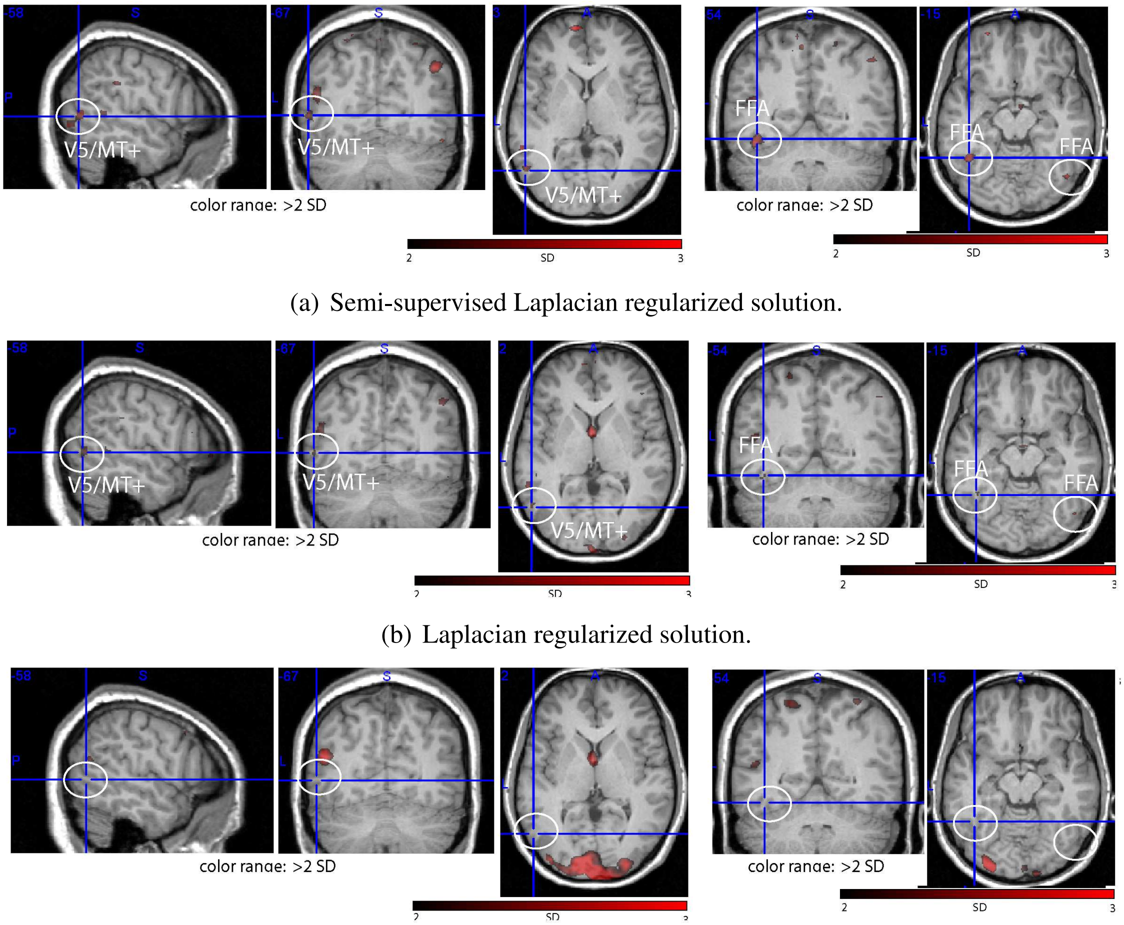


(a) Semi-supervised Laplacian regularized solution.

(b) Laplacian regularized solution.

(c) KCCA without Laplacian regularization.

Figure 6.5: *Motion: activation in the visual motion complex, area V5+/MT+. Same observation as in Figure 6.2.*



(a) Semi-supervised Laplacian regularized solution.

(b) Laplacian regularized solution.

(c) KCCA without Laplacian regularization.

Figure 6.6: *Multivariate -* Motion *and* Faces*: activations in the visual motion complex, area V5+/MT+ (left), and activation in the cortical region responsive to the visual perception of faces, the fusiform face area (FFA) (right). Same observation as in Figure 6.2.*

fMRI data. In the semi-supervised conditions (Experiment 3 as shown in Tables 6.1, 6.2 and 6.3) the additional data without correspondences is sufficiently close to the marginal distribution over $\mathcal{X}$ to improve results significantly, thus the additional data improves the results without any information about the correspondences of the data.

Additionally with the neuroscientific interpretation of the weights learned by the regression – in order to infer brain regions that are important during different types of visual processing – confirms previous studies (Bartels and Zeki (2004b); Bartels and Zeki (2004a); Bartels et al. (2007)). Figures 6.2 through 6.6 show slices taken through the anatomical image of one subject, with weight maps obtained from the different analyses of its functional data superimposed in red, wherein the maps were thresholded at 2 standard deviations in most cases, but had to be lowered in some cases to reveal any localized activity. We show examples of four of the single-variate labels for each of the three experiments, as well as one of the sets of multi-variate experiments. In the multi-variate label example, we show the same weight map but at different brain volume coordinates in order to visualize the expected brain activations for each of the lables involved. The maps corresponding well to the known functional anatomy, and to activations obtained in the previous regression studies of free-movie-viewing data Bartels and Zeki (2004a). Faces obtained high weights in the fusiform cortex (fusiform face area, FFA) (Figure 6.2); Human Bodies dorso-lateral and ventral parts within the lateral occipital cortex (extrastriate body area (EBA) and fusiform body area (FBA)) (Figure 6.11); Color obtained high weights in the medial fusiform cortex where human V4 is located (Figure 6.4). The spatial layout of the weights thus corresponds well to the previous literature, and indicates that some of the analyses applied here yield results that are neuroscientifically meaningful and that can identify distinct cortical regions involved in the distinct tasks. Semi-supervised Laplacian regularization worked well in that weight maps thresholded at $>$2SD show relatively well defined activity of the regions previously shown to be involved with the features. For other analyses, e.g. KCCA without Laplacian regularization, we had to reduce the threshold to 0.5 or 1 (faces and color in the single-variate cases, respectively) to obtain activity in the areas in question, and the maps show additional, unspecific activity as well.

6.4 Resting State Experiments

In the second set of experiments, the goal was to the assess how resting state data (Section 5.1.4) performs as a source of additional unlabeled data. Resting state data is even cheaper to obtain than active viewing data, in the sense that it has to be acquired in between active viewing recording sessions. Given this, its inclusion in the estimate of the manifold could even further boost inferential performance at a lesser cost.

This class of experiments is composed of the same experiments outlined for the natural/active viewing experiments in Section 6.3, but these are aimed at comparing resting state activity to active viewing data. As such, all forms of available data (described in Section 5.4) are utilized in these experiments, and have furthermore been acquired from five human volunteers. The hypotheses are that resting state activity is similar to natural visual processing data, and will thus perform just as well as a source of additional data in the semi-supervised learning framework.

The set of experiments consisted of:

(A) KCCA with Tikhonov regularization → labeled data only (supervised),

(B) KCCA with Tikhonov and Laplacian regularization → labeled data only (supervised),

- SSKCCA with Tikhonov and Laplacian regularization → labeled and *additional data* (semi-supervised):

  (C) resting state,

  (D) unlabeled active viewing data,

  (E) unlabeled active viewing data and resting state.

6.4.1 Results

The quantitative and qualitative results are presented in the first two sections below, and these results are summarized and discussed in the section to follow.

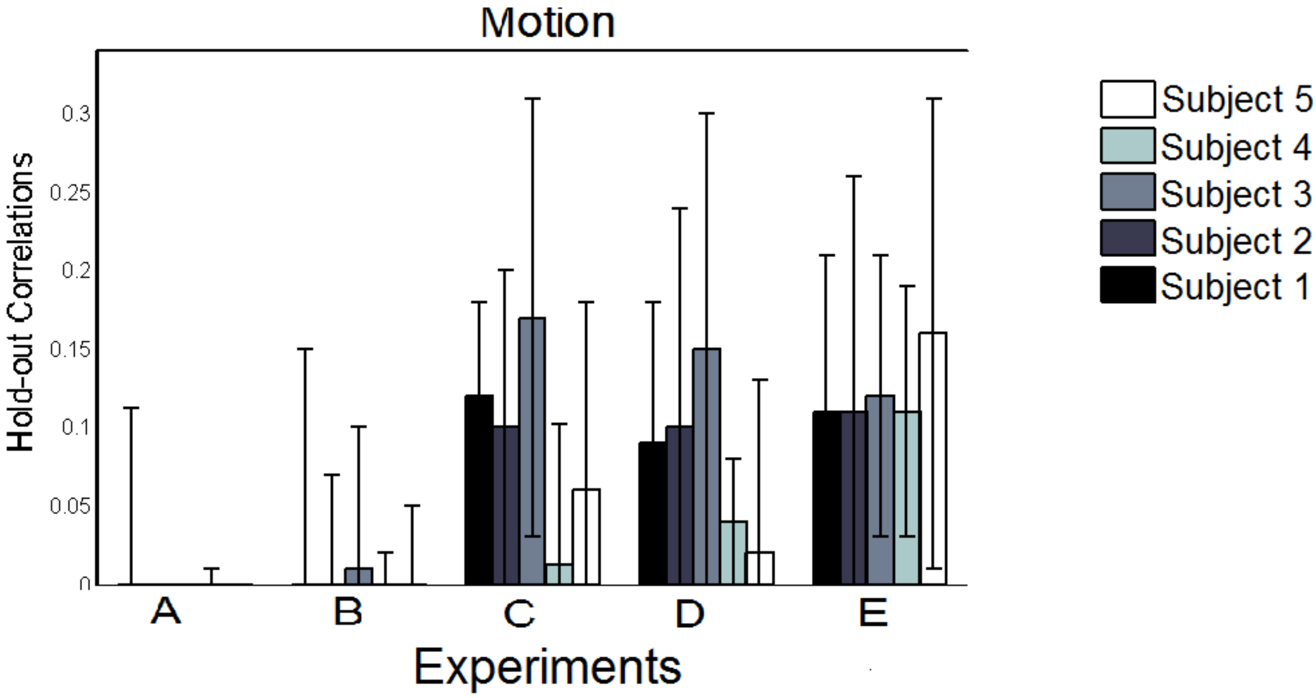
Quantitative Results

As in the first set of experiments (Section 6.3), we empirically evaluate the performance of the above KCCA variants via five-fold cross validation, and we model select both ε_x and λ_x regularization parameters with the criterion from Hardoon et al. (2004).

The results of these tables are depicted in Figure 6.7, Figure 6.8, and Figure 6.9.

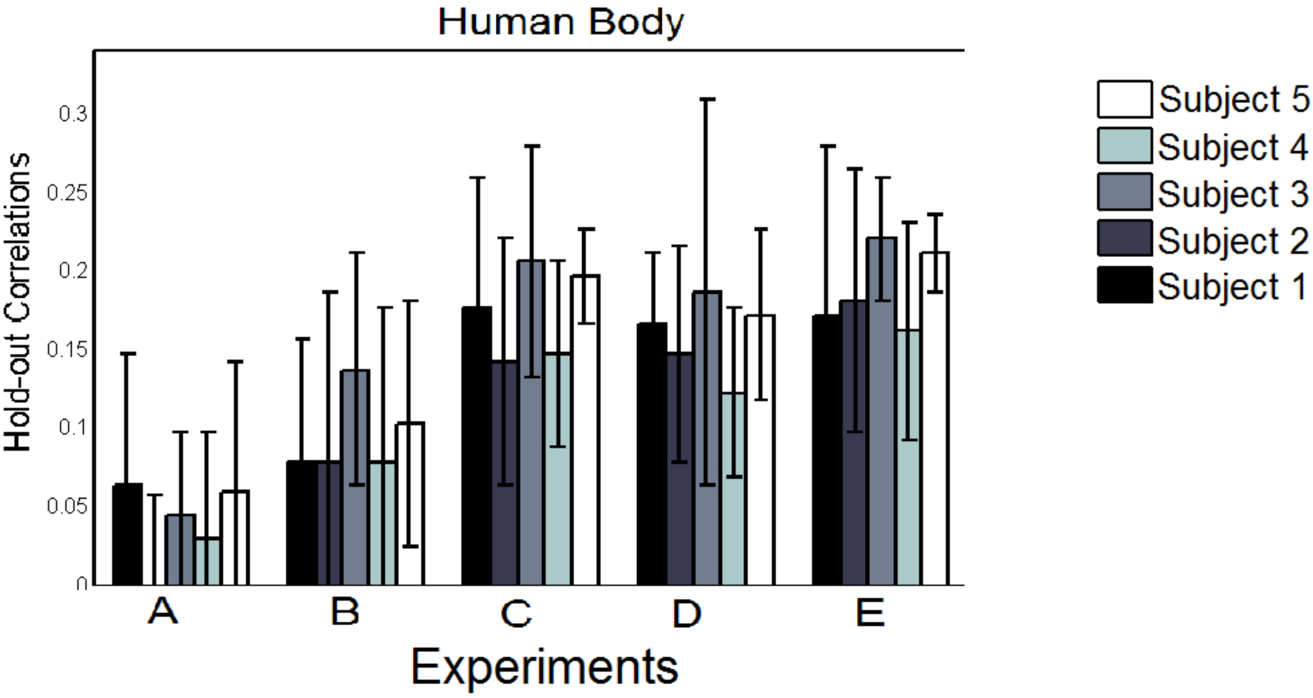
Qualitative Results

Again, as we used linear kernels as well in this set of experiments, we can visualize the learned projection vectors, the coefficients $\tilde{u}_x^1$ from f_x, as a map of the significant weights onto slices shown through single subjects. Figure 6.10 shows the weights for



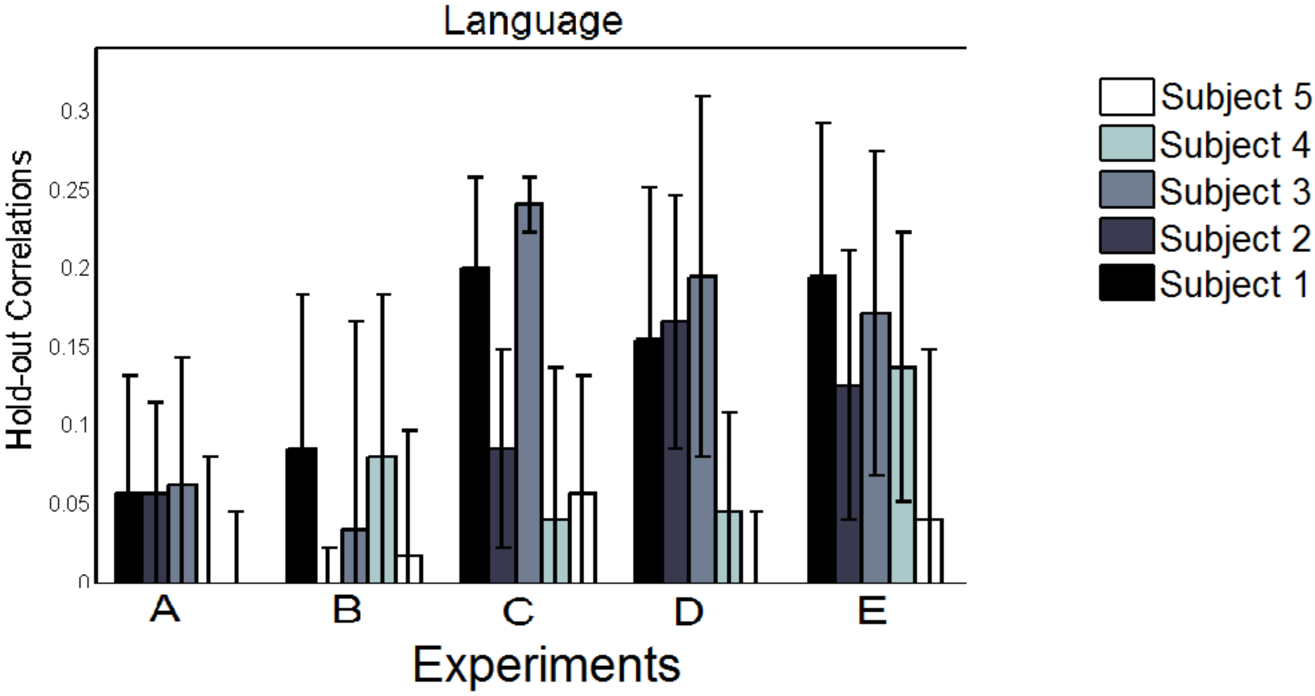
(a) Mean hold-out correlations for all experiments and all subjects in the label condition Motion.

Figure 6.7: *Experiment A: KCCA using only data for which correspondences are known, X, and Tikhonov regularization. Experiment B: Laplacian regularization where the graph Laplacian $\mathcal{L}$ is estimated using only data for which correspondences are known, X. Experiments C, D, E: full semi-supervised Laplacian regularization, $\mathcal{L}$ calculated using the augmented data matrix $\dot{X}$ computed from additional (C) resting state data, (D) unlabeled active viewing data, and (E) resting state and active viewing data. Semi-supervised Laplacian regularized KCCA in the C, D, and E experiments outperforms the two cases without semi-supervision (A, and B), and the performance of resting state data as an additional data source (C) performs comparatively well as the active viewing data conditions, (D) and (E).*



(a) Mean hold-out correlations for all experiments and all subjects in the label condition Human Bodies.

Figure 6.8: *Experiment A: KCCA using only data for which correspondences are known, X, and Tikhonov regularization. Experiment B: Laplacian regularization where the graph Laplacian $\mathcal{L}$ is estimated using only data for which correspondences are known, X. Experiments C, D, E: full semi-supervised Laplacian regularization, $\mathcal{L}$ calculated using the augmented data matrix $\dot{X}$ computed from additional (C) resting state data, (D) unlabeled active viewing data, and (E) resting state and active viewing data. Semi-supervised Laplacian regularized KCCA in the C, D, and E experiments outperforms the two cases without semi-supervision (A, and B), and the performance of resting state data as an additional data source (C) performs comparatively well as the active viewing data conditions, (D) and (E).*



(a) Mean hold-out correlations for all experiments and all subjects in the label condition Language.

Figure 6.9: *Experiment A: KCCA using only data for which correspondences are known, X, and Tikhonov regularization. Experiment B: Laplacian regularization where the graph Laplacian $\mathcal{L}$ is estimated using only data for which correspondences are known, X. Experiments C, D, E: full semi-supervised Laplacian regularization, $\mathcal{L}$ calculated using the augmented data matrix $\dot{X}$ computed from additional (C) resting state data, (D) unlabeled active viewing data, and (E) resting state and active viewing data. Semi-supervised Laplacian regularized KCCA in the C, D, and E experiments outperforms the two cases without semi-supervision (A, and B), and the performance of resting state data as an additional data source (C) performs comparatively well as the active viewing data conditions, (D) and (E).*

Table 6.4: *Mean holdout correlations for* motion *in the five subjects across all experiments. For a description of the experiments, see Section 6.4. In all cases, semi-supervision from resting state activity (Exp C) improves over regression using only fully labeled data (Exp A).*

	Sub 1	Sub 2	Sub 3	Sub 4	Sub 5
Exp A	-0.008 ± 0.12	-0.08 ± 0.07	-0.08 ± 0.04	-0.06 ± 0.07	-0.08 ± 0.08
Exp B	-0.02 ± 0.17	-0.03 ± 0.10	0.01 ± 0.09	-0.02 ± 0.04	-0.03 ± 0.08
Exp C	0.12 ± 0.06	0.10 ± 0.10	0.17 ± 0.14	0.012 ± 0.09	0.06 ± 0.12
Exp D	0.09 ± 0.09	0.10 ± 0.14	0.15 ± 0.15	0.04 ± 0.04	0.02 ± 0.11
Exp E	0.11 ± 0.10	0.11 ± 0.15	0.12 ± 0.09	0.11 ± 0.08	0.16 ± 0.15

Table 6.5: *Mean holdout correlations for* human body *in the five subjects across all experiments. For a description of the experiments, see Section 6.4. In all cases, semi-supervision from resting state activity (Exp C) improves over regression using only fully labeled data (Exp A).*

	Sub 1	Sub 2	Sub 3	Sub 4	Sub 5
Exp A	0.13 ± 0.17	-0.003 ± 0.12	0.09 ± 0.11	0.06 ± 0.14	0.12 ± 0.17
Exp B	0.16 ± 0.16	0.16 ± 0.22	0.28 ± 0.15	0.16 ± 0.20	0.21 ± 0.16
Exp C	0.36 ± 0.17	0.29 ± 0.16	0.42 ± 0.15	0.30 ± 0.12	0.40 ± 0.06
Exp D	0.34 ± 0.09	0.30 ± 0.14	0.38 ± 0.25	0.25 ± 0.11	0.35 ± 0.11
Exp E	0.35 ± 0.22	0.37 ± 0.17	0.45 ± 0.08	0.33 ± 0.14	0.43 ± 0.05

the *motion* variable, Figure 6.11 for the *human body* variable, and Figure 6.12 for the *language* variable.

Evaluation

The main trends to note from the quantitative results in Section 6.4.1 are that, in terms of additional data, resting state data performs as well as unlabeled active viewing data in a semi-supervised learning framework. Given that regularization with respect to their respective manifold estimates $\mathcal{L}$ yielded comparable results, this confirms our hypothesis

Table 6.6: *Mean holdout correlations for* language *in the five subjects across all experiments. For a description of the experiments, see Section 6.4. In all cases, semi-supervision from resting state activity (Exp C) improves over regression using only fully labeled data (Exp A).*

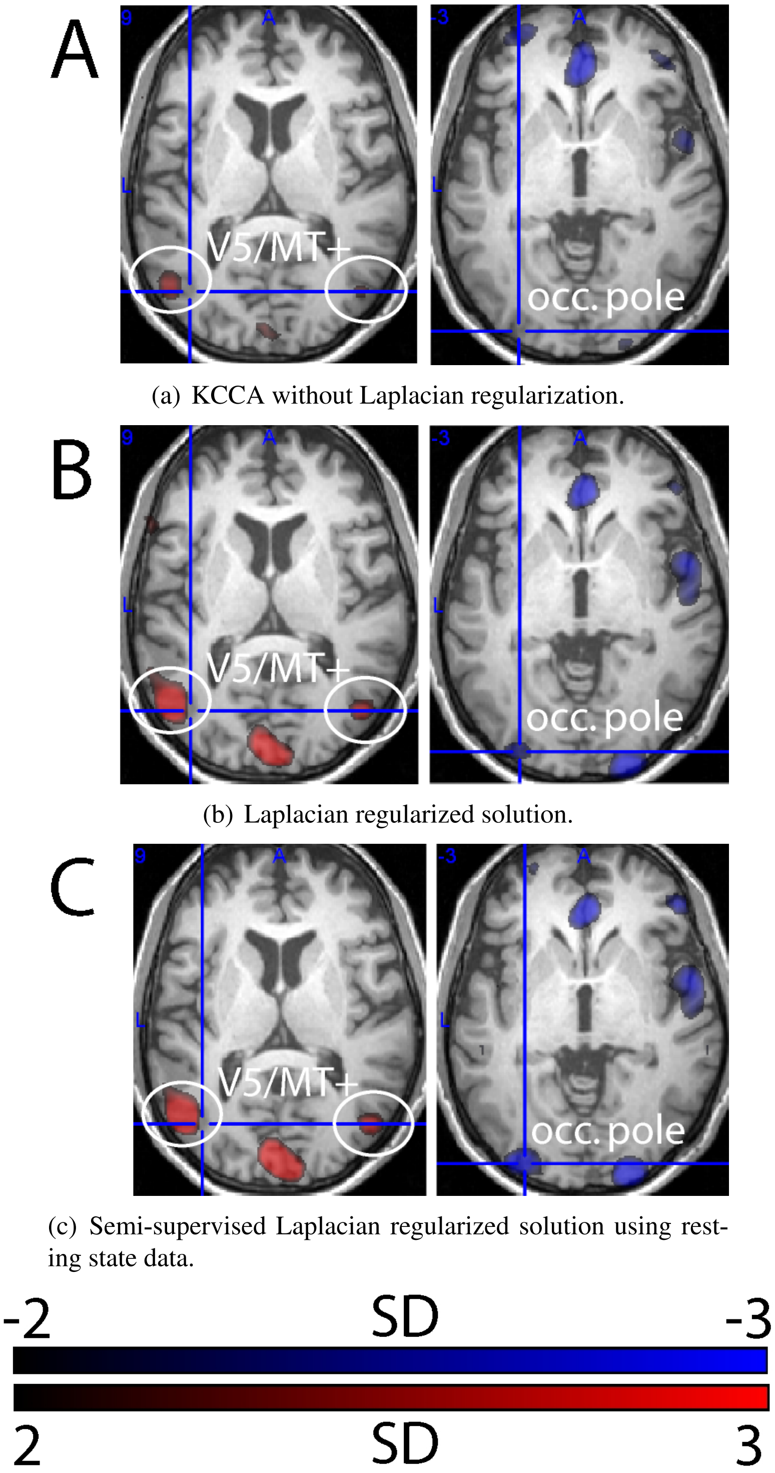
	Sub 1	Sub 2	Sub 3	Sub 4	Sub 5
Exp A	0.10 ± 0.13	0.10 ± 0.10	0.11 ± 0.14	-0.03 ± 0.17	-0.03 ± 0.11
Exp B	0.15 ± 0.17	-0.05 ± 0.09	0.06 ± 0.23	0.14 ± 0.18	0.03 ± 0.14
Exp C	0.35 ± 0.10	0.15 ± 0.11	0.42 ± 0.03	0.07 ± 0.17	0.10 ± 0.13
Exp D	0.27 ± 0.17	0.29 ± 0.14	0.34 ± 0.20	0.08 ± 0.11	-0.03 ± 0.11
Exp E	0.34 ± 0.17	0.22 ± 0.15	0.30 ± 0.18	0.24 ± 0.15	0.07 ± 0.19

that resting state data is similar in distributive structure to natural active viewing data. We note this comparison from the similarity in general improvement of Experiment C over Experiments A and B, across all subjects in each label condition, and that the improvement of (C) over (A) and (B) is comparable to the improvement in (D).

The feature-weight maps shown in Figures 6.10-6.12 were all in accord with established findings in neuroscience, in that distinct features such as visual motion, the perception of human bodies or of language correlated with activation of distinct brain regions, such as V5+/MT+, the lateral occipital complex (LOC) and the extrastriate body area (EBA), as well as regions of the STS and Wernickes area, respectively. These findings have now been established in studies using controlled stimuli, as well as those showing movie-clips to volunteers (Bartels and Zeki (2004b); Bartels et al. (2007); Hasson et al. (2004)).

Our results show that adding resting state data can indeed augment findings obtained in stimulus-inducing settings. This method may therefore be useful for the increasing number of imaging centers acquiring resting state data for completely different purposes, which may then be used to augment functional data, entirely free of cost in terms of scan time. An even more promising prospect however is that also the baseline or rest condition within stimulus-driven sessions may be used to augment the results obtained in the stimulus conditions. This may be especially valuable, since almost all imaging sessions contain baseline conditions, that are often not used for further analysis, but take up considerable amount of scan time.

Apart from the above, application-orientated considerations, our findings also provide new evidence that brain-states during rest which are difficult to characterize indeed resemble those during exposure to complex, natural stimulation. Our approach is therefore an extension of prior attempts to characterize the complex, rich, yet difficult to characterize brain activation during the absence of externally driven stimulation.

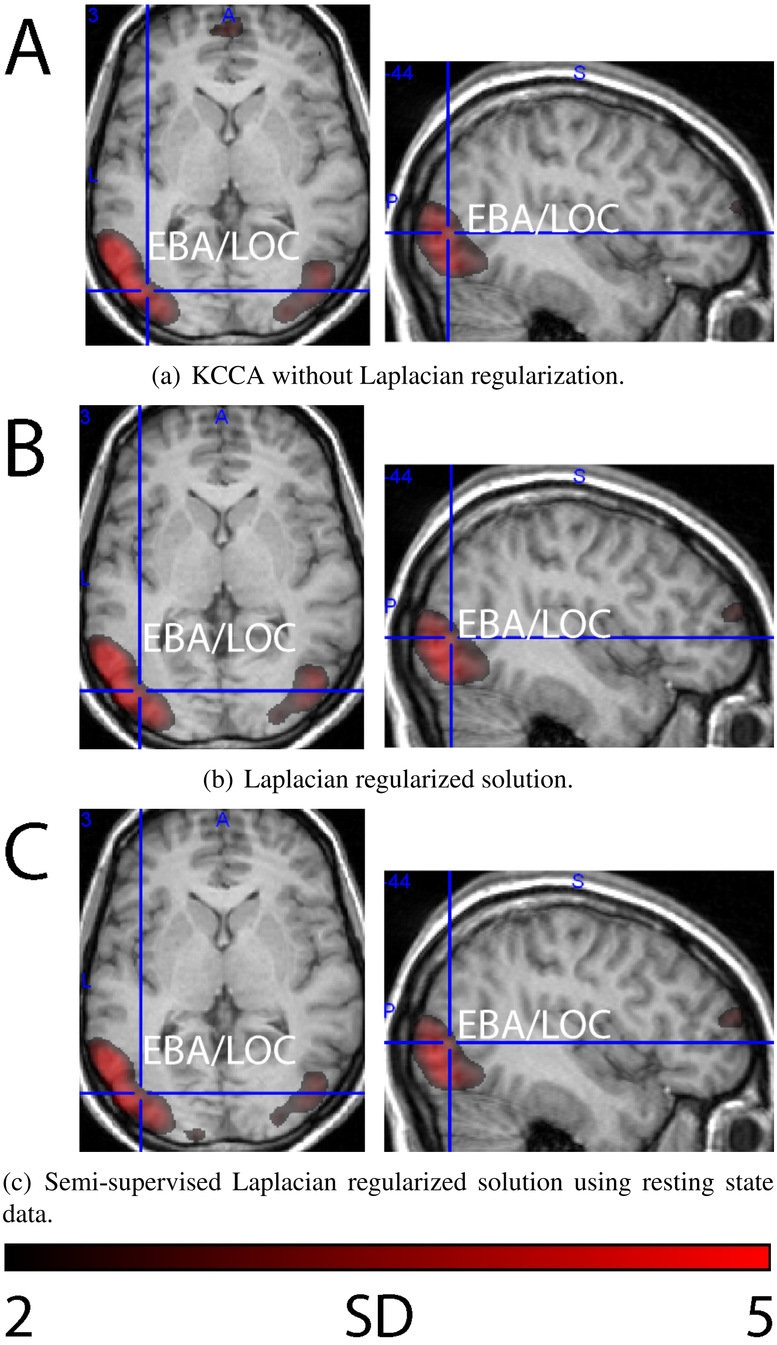


(a) KCCA without Laplacian regularization.

(b) Laplacian regularized solution.

(c) Semi-supervised Laplacian regularized solution using resting state data.

Figure 6.10: *Illustration of weight maps obtained for the visual motion feature in experiments A, B, and D. Transverse slices are shown through a single subjects T1-weighted structural image with superimposed weight-maps, colored in red for positive weights (left column), and colored in blue for negative weights (right column). The positive weight maps (left column) reveal the motion processing area V5/MT+, as well as posterior in the midline a part of peripheral early visual area V1 (not labeled). The negative weight maps reveal a reduction of BOLD signal in the occipital poles (the foveal representation of early visual areas V1-V3). Both results are in agreement with the findings reported in a prior study (Bartels et al. (2007)).*

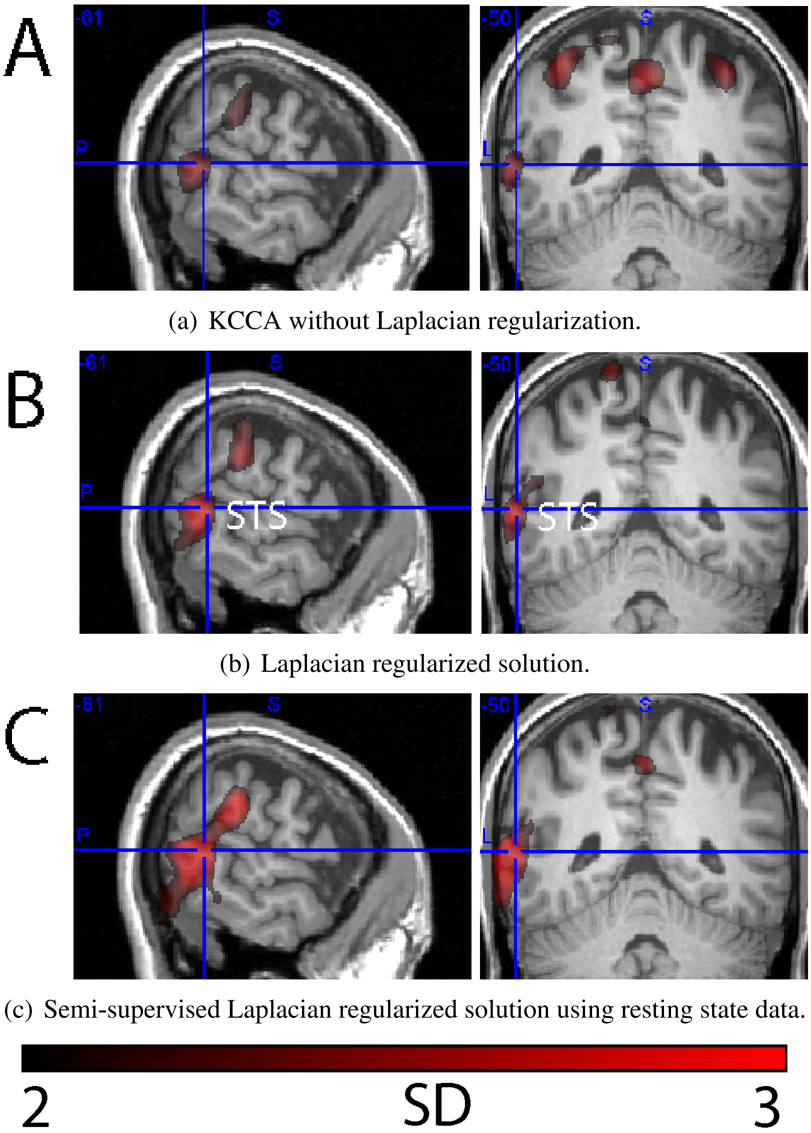


(a) KCCA without Laplacian regularization.

(b) Laplacian regularized solution.

(c) Semi-supervised Laplacian regularized solution using resting state data.

Figure 6.11: *Illustration of weight maps for the human body feature. Weight maps (in red) are show on transverse (left) and sagittal (right) brain sections of a single subject. Activity involves the object-responsive lateral occipital cortex (LOC) extending dorsally into region responsive to human bodies, dubbed extrastriate body area (EBA). The weights in all experiments are very strong for this feature (see colorbar), and nearly no difference in the extent of activation is visible across experiments.*



(a) KCCA without Laplacian regularization.

(b) Laplacian regularized solution.

(c) Semi-supervised Laplacian regularized solution using resting state data.

Figure 6.12: *Illustration of weight maps obtained for the language feature across the different experiments. Weight maps (in red) are superimposed on sagittal, coronal and transverse sections of a single subjects brain. The activation associated to this feature involved the superior temporal sulcus (STS), extending anteriorly to include parts of Wernickes speech processing area, and posterior and ventrally (increasing with experiments A, B and D) object-responsive region LOC, involved in analyzing facial features (in accord with the findings from Bartels and Zeki (2004b)).*

Chapter 7

Discussion

Digitalized signals of natural data is intrinsically of high dimensionality, as such, dimensionality reduction methods are crucial to the analysis of modern real-world data. In this thesis we have reviewed the core methods of dimensionality reduction (Chapter 2), with which one can learn a lower-dimensional linear representation of structure in high-dimensional data. We also saw how these linear algorithms can be generalized to account for more complex, nonlinear patterns via kernelization (Chapter 3). Finally, we learned how these methods can be extended in a semi-supervised framework for maximal usage of available data and greater generalizability of results (Chapter 4). The work in the thesis integrated these ideas and methods in the kernelized and semi-supervised dimensionality reduction technique of Canonical Correlation Analysis (CCA), called SSKCCA, and explored several variants of this algorithm with modern neuroscientific data, specifically with various forms of human fMRI data (Chapters 5 and 6), natural active viewing data and resting state data.

7.1 Conclusions

The two sets of experiments explored a novel approach to the analysis of human fMRI data acquired during a complex and natural viewing conditions (i.e. watching a movie), in order to infer the main regions of brain activity during a task. This approach was that of the recent dimensionality reduction technique called semi-supervised Laplacian regularized KCCA. The goal of these experiments was to make maximal use of all available fMRI data in the semi-supervised learning framework – even if there were (a) no known correspondences between stimulus and induced activity data and/or (b) no possible correspondences given a lack of stimulus – in order to reduce the need of the highly costly labeled data and reduce overfitting due to small sample sizes. This was approached, as mentioned via Laplacian regularization, which forms a non-parametric estimate of the underlying manifold structure of the data, or in other words geometrically estimates the structure of the marginal distribution, for which correspondences between stimulus and data are unneeded. Regularizing KCCA with respect to this structure yields projection directions in the brain data modality and stimulus modality where correlation is maximum and smooth along the manifold. KCCA was explored with two fully supervised variants (first with just Tikhonov regularization, then with Laplacian regularization using an estimate of the manifold structure using only the labeled data), and with several semi-supervised variants, which explored how different forms of additional (unlabeled) could be used to estimate the manifold and how they performed in Laplacian regularized KCCA.

The main messages of the experimental results presented in Sections 6.3.1 and 6.4.1 are as follows:

- The manifold assumption holds with the high-dimensional data acquired from fMRI studies.
- Laplacian regularized *semi-supervised* KCCA allows for successful *use of all available fMRI data*: by augmenting the manifold estimate with the less expensive forms of data which could not have been used in the supervised framework – the unlabeled active viewing data and/or resting state data – we could reduce the need for the expensive labeled data.
- Laplacian regularized SSKCCA *learned the expected regions*, or ground-truth, of brain activity corresponding to input stimuli as shown in previous neuroscientific studies. This shows that our method is learning what it should, and can indeed contribute to the neuroscientists' analysis of fMRI data.
- Laplacian regularized SSKCCA consistently generalized better to unseen data: regularizing the KCCA solution with respect to the augmented (unlabeled data or resting state data included) estimate yielded greater empirical hold-out correlations in every stimulus label condition and across all subjects.
- Finally, Laplacian regularization allows us to indeed reduce the cost of expensive labels for fMRI data, while improving performance of dimensionality reduction. Resting state data is a promising and inexpensive source of unlabeled data that can be used for this augmentation in the estimate of the manifold which results in this boost in empirical performance.

Appendix A

Appendix

A.1 Mathematical Foundations

A.1.1 Notation

Vectors, Matrices, and Mathematical Operations

$\mathbb{R}^d$	d-dimensional Euclidean space	(A.1)
X	capital letters denote matrices	(A.2)
x_{ij}	the entry in the i'th row, j'th column of X	(A.3)
x	lower-case letters denote (column) vectors	(A.4)
x_i	the i'th column vector of the respective matrix	(A.5)
I	identity matrix: square matrix, 1 along the diagonal, 0 else	(A.6)
x^T	transpose of vector x	(A.7)
$\|x\|$	Euclidean norm of vector x	(A.8)
X^{-1}	the inverse of matrix X	(A.9)
λ	eigenvalue	(A.10)
$x^T u$	projection of vector x onto vector u, when $\|u\| = 1$	(A.11)
$\max_x f(x)$	the value of x that leads to the maximum value of $f(x)$	(A.12)
$\sum_{n=1}^{n} a_i$	the sum from $i = 1$ to n, that is, $a_1 + a_2 + \cdots + a_n$	(A.13)

A.1.2 Linear Algebra

A d-dimensional column vector x and its transpose x^T can be written as

$$\textbf{Vectors} \qquad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix} \text{ and } x^T = (x_1, x_2, \dots, x_d) \tag{A.14}$$

An $d \times n$ rectangular matrix X and its transpose X^T are denoted as

$$\textbf{Matrices} \qquad X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{d1} & x_{d2} & \dots & x_{dn} \end{pmatrix} \tag{A.15}$$

and

$$X^T = \begin{pmatrix} x_{11} & x_{21} & \dots & x_{d1} \\ x_{12} & x_{22} & \dots & x_{d2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1n} & x_{2n} & \dots & x_{dn} \end{pmatrix} \tag{A.16}$$

The *inner product* of two vectors of the same dimensionality yields a scalar (and is symmetric)

$$\textbf{Inner Product} \qquad x^T w = \sum_{i=1}^{d} x_i w_i = w^T x. \tag{A.17}$$

The *Euclidean norm* (or length) of this vector is

$$\textbf{Euclidean Norm} \qquad ||x|| = \sqrt{x^T x}, \tag{A.18}$$

which we call 'normalized' when $||x|| = 1$. The inner product is a measure of collinearity of two vectors (given that the angle θ between two d-dimensional vectors is $cos\theta = \frac{x^T w}{||x|| ||w||}$), and thus a natural similarity measure between the vectors. Specifically, if $x^T w = 0$, then the two are orthogonal, whereas if $||x^T w|| = ||x|| ||w||$, they are collinear (for normalized vectors, collinear vectors would be $||x^T w|| = ||x|| ||w|| = 1$), or ”very similar”.

A set of vectors is *linearly independent* if no vector can be written as a linear combination of any of the other vectors. A set of d-dimensional linearly independent vectors

Linear Independence $$X = \{x_1, \ldots, x_n\} \in \mathbb{R}^d \qquad (A.19)$$

are said to span a d-dimensional vector space (i.e. $\mathbb{R}^d$), i.e. any vector in X can be written as a linear combination of the others spanning this space.

The *outter product* (or matrix product) of two vectors yields a matrix

Outter Product $$M = x^T w = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix} (w_1, w_2, \ldots, w_n) = \begin{pmatrix} x_1 w_1 & x_1 w_2 & \ldots & x_1 w_n \\ x_2 w_1 & x_2 w_2 & \ldots & x_2 w_n \\ \vdots & \vdots & \ddots & \vdots \\ x_d w_1 & x_d w_2 & \ldots & x_d w_n \end{pmatrix} \qquad (A.20)$$

Let $f(x)$ be a function of d variables in the vector $x^T = (x_1, \ldots, x_d)$. The derivative or gradient of $f(\cdot)$ with respect to x is computed component by component

Gradient of a function $$\nabla f(x) = grad\, f(x) = \frac{\partial f(x)}{\partial x} = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_d} \end{pmatrix} \qquad (A.21)$$

A.1.3 Probability Theory

The *mean* (first statistical moment) of the vector $x = (x_1, \ldots, x_n)$ is

Mean $$mean(x) = \frac{1}{n} \sum_{i=1}^{n} x_i. \qquad (A.22)$$

The *variance* (second statistical moment) of x is

Variance $$var(x) = \sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i \ - \ mean(x))^2. \qquad (A.23)$$

The *covariance* (a ”cross-moment”) of x and $y = (y_1, \ldots, y_n)$ is

Covariance $$covar(x) = \sigma_{xy} = \frac{1}{n} \sum_{i=1}^{n} (x_i \ - \ mean(x))(y_i \ - \ mean(x)), \qquad (A.24)$$

from which it follows that the *covariance matrix* C_{xx} is defined as the square matrix whose ijth element σ_{ij} is the covariance of x_i and x_j, and *cross-covariance matrix* C_{xy} contains in element σ_{ij} the covariance of x_i and y_j.

The *correlation coefficient* of x and y, defined as

Correlation Coefficient
$$corr(x) = \rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}, \tag{A.25}$$

is the normalized covariance of the two vectors. This will always yield a value ρ between -1 and $+1$, indicating the degree of negative or positive correlation, respectively, between the two vectors.

A.1.4 Lagrangian Optimization (KKT)

In this text, *Karush-Kuhn-Tucker conditions* are used to transform our optimization problems into generalized eigenvalue problems. Let x_0 be the position of the extremum value of a scalar-valued function $f(x)$ that we seek to find. $f(x)$ is subject to some constraint, and if we can express this in the form $g(x) = 0$, then we can find the extreme value, or optimum, of $f(x)$ by first finding the extremum x_0 with the following steps. First we form the *Lagrangian function*

Lagrangian Function
$$L(x, \lambda) = f(x) + \lambda g(x) \tag{A.26}$$

where λ is a scalar referred to as the *Lagrangian multiplier* that allows us to include the constraints on $f(x)$ in an 'unconstrained' way. We convert this constrained optimization problem into an unconstrained problem by finding the minimum value, taking the derivative of L with respect to x

$$\frac{\partial L(x, \lambda)}{\partial x} = \frac{\partial f(x)}{\partial x} + \lambda \frac{\partial g(x)}{\partial x} = 0, \tag{A.27}$$

through which we solve for λ. In general KKT conditions are necessary but not sufficient. However, for the Rayleigh quotient problems addressed in this thesis, the maximum eigenvalue also corresponds to the global maximum of the optimization problem. In other words, the solution provides the x position of the extremum (x_0), and via substitution of this value we find the optimum (extreme value) of $f(\cdot)$ under the given constraint(s).

List of Figures

List of Tables

Bibliography

Andersen, T. L., Martinez, T. R., 2001. Dmp3: A dynamic multilayer perceptron construction algorithm. Int. J. Neural Syst. 11 (2), 145–165.

Bach, F. R., Jordan, M. I., 2002. Kernel Independent Component Analysis. JMLR 3, 1–48.

Bartels, A., Zeki, S., 2004a. The chronoarchitecture of the human brain–natural viewing conditions reveal a time-based anatomy of the brain. NeuroImage 22 (1), 419 – 433.

Bartels, A., Zeki, S., 02/01/ 2004b. Functional brain mapping during free viewing of natural scenes. Human Brain Mapping 21 (2), 75–85.

Bartels, A., Zeki, S., Logothetis, N. K., July 2007. Natural vision reveals regional specialization to local motion and to contrast-invariant, global flow in the human brain. Cereb. Cortex, bhm107+.

Belhumeur, P., Hespanha, J., Kriegman, D., Jul 1997. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. Pattern Analysis and Machine Intelligence, IEEE Transactions on 19 (7), 711–720.

Belkin, M., Niyogi, P., Sindhwani, V., 2006. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. JMLR 7, 2399–2434.

Bellman, R., 1961. Adaptive Control Processes. Princeton University Press.

Bishop, C. M., 2006. Pattern Recognition and Machine Learning. Springer Science.

Biswal, B., Yetkin, F. Z., Haughton, V. M., Hyde, J. S., October 1995. Functional connectivity in the motor cortex of resting human brain using echo-planar mri. Magnetic resonance in medicine 34 (4), 537–541.

Blaschko, M., Shelton, J., Bartels, A., 2009. Augmenting Feature-driven fMRI Analyses: Semi-supervised learning and resting state activity. In: NIPS.

Blaschko, M., Shelton, J., Bartels, A., Lampert, C., Gretton, A., submitted. Semi-supervised kernel canonical correlation analysis with application to human fmri. Pattern Recognition Letters.

Blaschko, M. B., Lampert, C. H., 2008. Correlational Spectral Clustering. In: CVPR.

Blaschko, M. B., Lampert, C. H., Gretton, A., 2008. Semi-supervised laplacian regularization of kernel canonical correlation analysis. In: ECML PKDD '08: Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I. Springer-Verlag, Berlin, Heidelberg, pp. 133–145.

Borga, M., Borga, C. M., 1998. Learning multidimensional signal processing. Tech. rep., Linkping University, Sweden.

Bousquet, O., von Luxburg, U., Rätsch, G. (Eds.), 2004. Advanced Lectures on Machine Learning, ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures. Vol. 3176 of Lecture Notes in Computer Science. Springer.

Burges, C., 2004. Some notes on applied mathematics for machine learning. Microsoft Technical Report (MSR-TR-2004-56).

Cai, D., He, X., Han, J., 2007. Semi-supervised discriminant analysis. In: ICCV.

Chapelle, O., Schölkopf, B., Zien, A. (Eds.), 2006. Semi-Supervised Learning. MIT Press, Cambridge, MA.

Duda, R. O., Hart, P. E., Stork, D. G., 2001. Pattern Classification, 2nd Edition. Wiley, New York.

Friston, K., Ashburner, J., Kiebel, S., Nichols, T., Penny, W. (Eds.), 2007. Statistical Parametric Mapping: The Analysis of Functional Brain Images. Academic Press.

Hardoon, D. R., Mourão-Miranda, J., Brammer, M., Shawe-Taylor, J., 2007. Unsupervised Analysis of fMRI Data Using Kernel Canonical Correlation. NeuroImage 37 (4), 1250–1259.

Hardoon, D. R., Szedmák, S., Shawe-Taylor, J. R., 2004. Canonical Correlation Analysis: An Overview with Application to Learning Methods. Neural Computation 16 (12), 2639–2664.

Hasson, U., Nir, Y., Levy, I., Fuhrmann, G., Malach, R., 2004. Intersubject Synchronization of Cortical Activity During Natural Vision. Science 303 (5664), 1634–1640.

Hasson, U., Yang, E., Vallines, I., Heeger, D. J., Rubin, N., March 2008. A hierarchy of temporal receptive windows in human cortex. J. Neurosci. 28 (10), 2539–2550.

Hein, M., Audibert, J.-Y., von Luxburg, U., 2006. Graph laplacians and their convergence on random neighborhood graphs. CoRR.

Hotelling, H., 1936. Relations Between Two Sets of Variates. Biometrika 28, 321–377.

Lai, P., Fyfe, C., 2000. Kernel and nonlinear canonical correlation analysis. Int. J. Neural Syst. 10 (5), 365–377.

Lampert, C. H., 2009. Kernel methods in computer vision. Foundations and Trends in Computer Graphics and Vision 4, 193–285.

Leurgans, S. E., Moyeed, R. A., Silverman, B. W., 1993. Canonical correlation analysis when the data are curves. Journal of the Royal Statistical Society, Series B (Methodological) 55 (3), 725–740.

Schölkopf, B., Smola, A. J., December 2002. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. The MIT Press.

Shelton, J., Blaschko, M., Bartels, A., 2009a. Semi-supervised subspace analysis of human functional magnetic resonance imaging data. Max Planck Institute Technical Report (185).

Shelton, J. A., Blaschko, M. B., Lampert, C. H., Bartels, A., 2009b. Semi-supervised subspace analysis of human fmri data. Berlin Brain Computer Infercace Workshop.

Sun, L., Ji, S., Ye, J., 2008. A least squares formulation for canonical correlation analysis. 307, 1024–1031.

Sun, L., Ji, S., Yu, S., Ye, J., 2009. On the equivalence between canonical correlation analysis and orthonormalized partial least squares., 1230–1235.

Tikhonov, A. N., 1963. Solution of incorrectly formulated problems and the regularization method. Soviet Math. Dokl.

Tikhonov, A. N., Arsenin, V. Y., 1977. Solution of Ill-posed Problems. John Wiley & Sons.

Ulmer, S., Jansen, O., 2010. fMRI: Basics and Clinical Applications.

von Luxburg, U., 2007. A Tutorial on Spectral Clustering. Statistics and Computing 17 (4), 395–416.

von Luxburg, U., Bousquet, O., Belkin, M., 2004. On the convergence of spectral clustering on random samples: The normalized case. In: Shawe-Taylor, J., Singer, Y. (Eds.), COLT. Vol. 3120 of Lecture Notes in Computer Science. Springer, pp. 457–471.

Zhou, D., Schölkopf, B., 11 2006. Discrete regularization. In: Chapelle, O., Schölkopf, B., Zien, A. (Eds.), Semi-supervised learning. Adaptive computation and machine learning. MIT Press, Cambridge, Mass., USA, pp. 221–232.

Zhu, X., 2005. Semi-supervised learning literature survey. Tech. Rep. 1530, Computer Sciences, University of Wisconsin-Madison.