Dimension Reduction: A Guided Tour

Christopher J.C. Burges[1]

[1] *Microsoft Research, One Microsoft Way, Redmond, WA 98052-6399, USA.*

MSR Tech Report MSR-TR-2009-2013

Abstract

We give a tutorial overview of several geometric methods for dimension reduction. We divide the methods into projective methods and methods that model the manifold on which the data lies. For projective methods, we review projection pursuit, principal component analysis (PCA), kernel PCA, probabilistic PCA, canonical correlation analysis, oriented PCA, and several techniques for sufficient dimension reduction. For the manifold methods, we review multidimensional scaling (MDS), landmark MDS, Isomap, locally linear embedding, Laplacian eigenmaps and spectral clustering. The Nyström method, which links several of the manifold algorithms, is also reviewed. The goal is to provide a self-contained overview of key concepts underlying many of these algorithms, and to give pointers for further reading.

Contents

1

Introduction

Dimension reduction[1] is the mapping of data to a lower dimensional space such that uninformative variance in the data is discarded, or such that a subspace in which the data lives is detected. Dimension reduction has a long history as a method for data visualization, and for extracting key low dimensional features (for example, the 2-dimensional orientation of an object, from its high dimensional image representation). In some cases the desired low dimensional features depend on the task at hand. Apart from teaching us about the data, dimension reduction can lead us to better models for inference. The need for dimension reduction also arises for other pressing reasons. [Stone, 1982] showed that, under certain regularity assumptions (including that the samples be IID), the optimal rate of convergence[2] for nonparametric

[1] We follow both the lead of the statistics community and the spirit of the paper to reduce "dimensionality reduction" and "dimensional reduction" to "dimension reduction".

[2] For convenience we reproduce Stone's definitions [Stone, 1982]. A "rate of convergence" is defined as a sequence of numbers, indexed by sample size. Let θ be the unknown regression function, Θ the collection of functions to which θ belongs, $\hat{T}_n$ an estimator of θ using n samples, and $\{b_n\}$ a sequence of positive constants. Then $\{b_n\}$ is called a lower rate of convergence if there exists $c > 0$ such that $\lim_n \inf_{\hat{T}_n} \sup_{\Theta} P(\|\hat{T}_n - \theta\| \geq cb_n) = 1$, and it is called an achievable rate of convergence if there is a sequence of estimators $\{\hat{T}_n\}$

regression varies as $m^{-p/(2p+d)}$, where m is the sample size, the data lies in $\mathcal{R}^d$, and where the regression function is assumed to be p times differentiable. We can get a very rough idea of the impact of sample size on the rate of convergence as follows. Consider a particular point in the sequence of values corresponding to the optimal rate of convergence: $m = 10,000$ samples, for $p = 2$ and $d = 10$. Suppose that d is increased to 20; what number of samples in the new sequence gives the same value? The answer is approximately 10 million. If our data lie (approximately) on a low dimensional manifold $\mathcal{L}$ that happens to be embedded in a high dimensional manifold $\mathcal{H}$, then modeling the data directly in $\mathcal{L}$ rather than in $\mathcal{H}$ may turn an infeasible problem into a feasible one.

The purpose of this review is to describe the mathematics and key ideas underlying the methods, and to provide some links to the literature for those interested in pursuing a topic further[3]. The subject of dimension reduction is vast, so we use the following criterion to limit the discussion: we restrict our attention to the case where the inferred feature values are continuous. The observables, on the other hand, may be continuous or discrete. Thus this review does not address clustering methods, or, for example, feature selection for discrete data, such as text. Furthermore implementation details, and important theoretical details such as consistency and rates of convergence of sample quantities to their population values, although important, are not discussed. Regarding notation: vectors are denoted by boldface, whereas components are denoted by x_a, or by $(\mathbf{x}_i)_a$ for the a'th component of the i'th vector. Random variables are denoted by upper case; we use $E[X|y]$ as shorthand for the function $E[X|Y = y]$, in contrast to the random variable $E[X|Y]$. Following [Horn and Johnson, 1985], the set of p by q matrices is denoted M_{pq}, the set of (square) p by p matrices by M_p, and the set of symmetric p by p matrices by S_p (all matrices considered are real). $\mathbf{e}$ with no subscript is used to denote the vector of all ones; on the other hand $\mathbf{e}_a$ denotes the a'th eigenvector. We denote sample

and $c > 0$ such that $\lim_n \sup_\Theta P(\|\hat{T}_n - \theta\| \geq cb_n) = 0$; $\{b_n\}$ is called an optimal rate of convergence if it is both a lower rate of convergence and an achievable rate of convergence. Here the $\inf_{\hat{T}_n}$ is over all possible estimators $\hat{T}_n$.

[3] This paper is a revised and extended version of [Burges, 2005].

size by m, and dimension usually by d or d', with typically $d' \ll d$. δ_{ij} is the Kronecker delta (the ij'th component of the unit matrix). We generally reserve indices i, j, to index vectors and a, b to index dimension.

We place dimension reduction techniques into two broad categories: methods that rely on projections (Chapter 2 2) and methods that attempt to model the manifold on which the data lies (Chapter 3 3). Chapter 2 gives a detailed description of principal component analysis; apart from its intrinsic usefulness, PCA is interesting because it serves as a starting point for many modern algorithms, some of which (kernel PCA, probabilistic PCA, and oriented PCA) are also described here. However it has clear limitations: it is easy to find even low dimensional examples where the PCA directions are far from optimal for feature extraction [Duda and Hart, 1973], and PCA ignores correlations in the data that are higher than second order. We end Chapter 2 with a brief look at projective methods for dimension reduction of *labeled* data: sliced inverse regression, and kernel dimension reduction. Chapter 3 starts with an overview of the Nyström method, which can be used to extend, and link, several of the algorithms described in this paper. We then examine some methods for dimension reduction which assume that the data lie on a low dimensional manifold embedded in a high dimensional space, namely locally linear embedding, multidimensional scaling, Isomap, Laplacian eigenmaps, and spectral clustering.

2

Projective Methods

If dimension reduction is so desirable, how should we go about it? Perhaps the simplest approach is to attempt to find low dimensional *projections* that extract useful information from the data, by maximizing a suitable objective function. This is the idea of projection pursuit [Friedman and Tukey, 1974]. The name 'pursuit' arises from the iterative version, where the currently optimal projection is found in light of previously found projections (in fact originally this was done manually[1]). Apart from handling high dimensional data, projection pursuit methods can be robust to noisy or irrelevant features [Huber, 1985], and have been applied to regression [Friedman and Stuetzle, 1981], where the regression is expressed as a sum of 'ridge functions' (functions of the one dimensional projections) and at each iteration the projection is chosen to minimize the residuals; to classification; and to density estimation [Friedman et al., 1984]. How are the interesting directions found? One approach is to search for projections such that the projected data departs from normality [Huber, 1985]. One might think that, since a distribution is normal if and only if all of its one dimen-

[1] See J.H. Friedman's interesting response to [Huber, 1985] in the same issue.

sional projections are normal, if the least normal projection of some dataset is still approximately normal, then the dataset is also necessarily approximately normal, but this is not true; Diaconis and Freedman have shown that most projections of high dimensional data are approximately normal [Diaconis and Freedman, 1984] (see also below). Given this, finding projections along which the density departs from normality, if such projections exist, should be a good exploratory first step.

The sword of Diaconis and Freedman cuts both ways, however. If most projections of most high dimensional datasets are approximately normal, perhaps projections are not always the best way to find low dimensional representations. Let's review their results in a little more detail. The main result can be stated informally as follows: consider a model where the data, the dimension d, and the sample size m depend on some underlying parameter ν, such that as ν tends to infinity, so do m and d. Suppose that as ν tends to infinity, the fraction of vectors which are not approximately the same length tends to zero, and suppose further that under the same conditions, the fraction of pairs of vectors which are not approximately orthogonal to each other also tends to zero[2]. Then ([Diaconis and Freedman, 1984], Theorem 1.1) the empirical distribution of the projections along any given unit direction tends to $N(0, \sigma^2)$ weakly in probability[3]. However, if the conditions are not fulfilled, as for some long-tailed distributions, then the opposite result can hold - that is, most projections are *not* normal (for example, most projections of Cauchy distributed data[4] will be Cauchy [Diaconis and Freedman, 1984]).

As a concrete example, consider data uniformly distributed over the unit $n+1$-sphere $\mathcal{S}^{n+1}$ for odd[5] n. Let's compute the density projected along any line $\mathcal{I}$ passing through the origin. By symmetry, the result will be independent of the direction we choose. If the distance along

[2] More formally, the conditions are: for σ^2 positive and finite, and for any positive ϵ, $(1/m)\text{card}\{j \leq m : |\|\mathbf{x}_j\|^2 - \sigma^2 d| > \epsilon d\} \rightarrow 0$ and $(1/m^2)\text{card}\{1 \leq j, k \leq m : |\mathbf{x}_j \cdot \mathbf{x}_k| > \epsilon d\} \rightarrow 0$ [Diaconis and Freedman, 1984].

[3] Some authors refer to convergence 'weakly in probability' simply as convergence in probability. A sequence X_n of random variables is said to converge in probability to a random variable X if $lim_{n \rightarrow \infty} P(|X_n - X| > \epsilon) = 0$ for all $\epsilon > 0$ [Grimmet and Stirzaker, 2001].

[4] The Cauchy distribution in one dimension has density $c/(c^2 + x^2)$ for constant c.

[5] The story for even n is similar but the formulae are slightly different.

the projection is parameterized by $\xi \equiv \cos\theta$, where θ is the angle between $\mathcal{I}$ and the line from the origin to a point on the sphere, then the density at ξ is proportional to the volume of an n-sphere of radius $\sin\theta$: $\rho(\xi) = C(1-\xi^2)^{\frac{n-1}{2}}$. Requiring that $\int_{-1}^{1}\rho(\xi)d\xi = 1$ gives the constant C:

$$C = 2^{-\frac{1}{2}(n+1)}\frac{n!!}{(\frac{1}{2}(n-1))!} \tag{2.1}$$

Let's plot this density and compare against a one dimensional Gaussian density fitted using maximum likelihood. For that we just need the variance, which can be computed analytically: $\sigma^2 = \frac{1}{n+2}$, and the mean, which is zero. Figure 2.1 shows the result for the 20-sphere. Although data uniformly distributed on $\mathcal{S}^{20}$ is far from Gaussian, its projection along any direction is close to Gaussian for all such directions, and we cannot hope to uncover such structure using one dimensional projections.

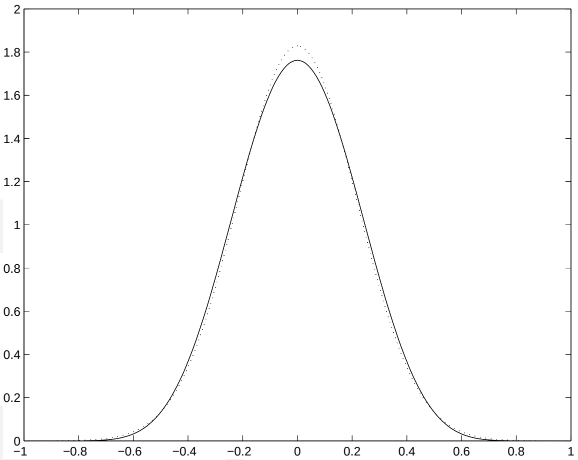

Fig. 2.1 Dotted line: a Gaussian with zero mean and variance 1/21. Solid line: the density projected from data distributed uniformly over the 20-sphere, to any line passing through the origin.

The notion of searching for non-normality, which is at the heart of projection pursuit (the goal of which is dimension reduction), is also a key idea underlying independent component analysis (ICA) [Hyvärinen et al., 2001]. ICA views the data as being generated by a mixture of unknown latent variables, and although typically the number of latent variables is assumed to equal the dimension of the data, the method has parallels with dimension reduction, so we briefly describe it here. ICA searches for projections such that the probability distributions of the data along those projections are statistically independent. Consider for example the case of two speakers speaking into two microphones, where each microphone captures sound from both speakers. The microphone signals may be written $\mathbf{y} = A\mathbf{x}$, $\mathbf{x}$, $\mathbf{y} \in R^2$, where the components of $\mathbf{x}$ are the (assumed statistically independent and zero mean) signals from each individual speaker, and where A is a fixed two dimensional mixing matrix. In principle, we could separate out the source signals by finding A and inverting it. However, both A and $\mathbf{x}$ are unknown here, and any invertible scaling of each component of $\mathbf{x}$, followed by any permutation of the components of the rescaled $\mathbf{x}$ (the net result of which is another pair of statistically independent variables) can be compensated by redefining A. We can remove the scaling degrees of freedom from the problem by whitening the data $\mathbf{y}$ and then assuming that A is a rotation matrix, which amounts to choosing a coordinate system in which $\mathbf{x}$ is white (which, since the x_i are independent and zero mean, is equivalent to just rescaling the x_i). Note that this also means that if $\mathbf{x}$ happens to be normally distributed, then ICA fails, since A can then be any orthogonal matrix (since any orthogonal matrix applied to independent, unit variance Gaussian variables results in independent, unit variance Gaussian variables). To give nontrivial results, ICA therefore requires that the original signals be non-Gaussian (or more precisely, that at most one is Gaussian distributed), and in fact it turns out that finding the maximally non-Gaussian component (under the assumptions that the $\mathbf{x}$ are IID, zero mean, and unit variance) will yield an independent component [Hyvärinen et al., 2001]. ICA components may also be found by searching for components with minimum mutual information, since zero mutual information corresponds to statistical independence. Such functions - whose optimization leads to the desired independent components - are called contrast functions. [Bach and Jordan, 2002] generalize ICA by proposing contrast functions based on canonical correlation analysis (CCA) in Reproducing Kernel Hilbert Spaces (RKHSs); we will encounter CCA, and RKHS's used in similar ways, below.

2.1 Principal Components Analysis (PCA)

2.1.1 PCA: Finding an Informative Direction

Given data $\mathbf{x}_i \in \mathcal{R}^d$, $i = 1, \cdots, m$, suppose you'd like to find a direction $\mathbf{v} \in \mathcal{R}^d$ for which the projection $\mathbf{x}_i \cdot \mathbf{v}$ gives a good one dimensional representation of your original data: that is, informally, the act of projecting loses as little information about your expensively-gathered data as possible (we will examine the information theoretic view of this below). Suppose that unbeknownst to you, your data in fact lies along a line $\mathcal{I}$ embedded in $\mathcal{R}^d$, that is, $\mathbf{x}_i = \boldsymbol{\mu} + \theta_i \mathbf{n}$, where $\boldsymbol{\mu}$ is the sample mean[6], $\theta_i \in \mathcal{R}$, $\sum_i \theta_i = 0$, and $\mathbf{n} \in \mathcal{R}^d$ has unit length. The sample variance of the projection along $\mathbf{n}$ is then[7]

$$v_n \equiv \frac{1}{m}\sum_{i=1}^{m}((\mathbf{x}_i - \boldsymbol{\mu}) \cdot \mathbf{n})^2 = \frac{1}{m}\sum_{i=1}^{m}\theta_i^2 \tag{2.2}$$

and that along some other unit direction $\mathbf{n}'$ is

$$v_n' \equiv \frac{1}{m}\sum_{i=1}^{m}((\mathbf{x}_i - \boldsymbol{\mu}) \cdot \mathbf{n}')^2 = \frac{1}{m}\sum_{i=1}^{m}\theta_i^2(\mathbf{n} \cdot \mathbf{n}')^2 \tag{2.3}$$

Since $(\mathbf{n} \cdot \mathbf{n}')^2 = \cos^2\phi$, where ϕ is the angle between $\mathbf{n}$ and $\mathbf{n}'$, we see that the projected variance is maximized if and only if $\mathbf{n} = \pm\mathbf{n}'$. Hence in this case, finding the projection for which the projected variance is maximized gives you the direction you are looking for, namely $\mathbf{n}$, *regardless of the distribution of the data along* $\mathbf{n}$, as long as the data has finite variance. You would then quickly find that the variance along all directions orthogonal to $\mathbf{n}$ is zero, and conclude that your data in fact lies along a one dimensional manifold embedded in $\mathcal{R}^d$. This is one of several basic results of PCA that hold for arbitrary distributions, as we shall see.

Even if the underlying physical process generates data that ideally lies along $\mathcal{I}$, noise will usually modify the data at various stages up to

[6] Note that if all x_i lie on a given line then so does μ.

[7] When the choice is immaterial to the argument, we use denominator m (sample viewed as the whole population) rather than $m - 1$ (unbiased estimator of population variance).

and including the measurements themselves, and so your data will very likely not lie exactly along $\mathcal{I}$. If the overall noise is much smaller than the signal, it makes sense to try to find $\mathcal{I}$ by searching for that projection along which the projected data has maximal variance. If instead your data lies in a two (or higher) dimensional subspace, the above argument can be repeated, picking off the highest variance directions in turn. Let's see how that works.

2.1.2 PCA: Ordering by Variance

We have seen that directions of maximum variance can be interesting, but how can we find them? From here on, unless otherwise stated, we allow the x_i to be arbitrarily distributed. The sample variance along an arbitrary unit vector $\mathbf{n}$ is $\mathbf{n}^T C\mathbf{n}$ where C is the sample covariance matrix. Since C is positive semidefinite, its eigenvalues are positive or zero; let us choose the indexing such that the (unit norm) eigenvectors $\mathbf{e}_a,\ a = 1, \dots, d$ are arranged in order of decreasing size of the corresponding eigenvalues λ_a. Since the $\{\mathbf{e}_a\}$ span the space (or can be so chosen, if several share the same eigenvalue), we can expand any $\mathbf{n}$ in terms of them: $\mathbf{n} = \sum_{a=1}^{d} \alpha_a \mathbf{e}_a$, and we would like to find the α_a that maximize $\mathbf{n}^T C\mathbf{n} = \mathbf{n}^T \sum_a \alpha_a C\mathbf{e}_a = \sum_a \lambda_a \alpha_a^2$, subject to $\sum_a \alpha_a^2 = 1$ (to give unit normed $\mathbf{n}$). This is just a convex combination of the λ's, and since a convex combination of any set of numbers is maximized by taking the largest, the optimal $\mathbf{n}$ is just $\mathbf{e}_1$, the principal eigenvector (or any one of the principal eigenvectors, if the principal eigenvalue has geometric multiplicity greater than one), and furthermore, the sample variance of the projection of the data along $\mathbf{n}$ is then just λ_1.

The above construction captures the variance of the data along the direction $\mathbf{n}$. To characterize the remaining variance of the data, let's find that direction $\mathbf{m}$ which is both orthogonal to $\mathbf{n}$, and along which the projected data again has maximum variance. Since the eigenvectors of C form an orthonormal basis (or can be so chosen), we can expand $\mathbf{m}$ in the subspace $\mathcal{R}^{d-1}$ orthogonal to $\mathbf{n}$ as $\mathbf{m} = \sum_{a=2}^{d} \beta_a \mathbf{e}_a$. Just as above, we wish to find the β_a that maximize $\mathbf{m}^T C\mathbf{m} = \sum_{a=2}^{d} \lambda_a \beta_a^2$, subject to $\sum_{a=2}^{d} \beta_a^2 = 1$, and by the same argument, the desired direction is given by the (or any) remaining eigenvector with largest eigenvalue, and the corresponding variance is just that eigenvalue. Repeating this argument gives d orthogonal directions, in order of monotonically decreasing projected variance. PCA for feature extraction thus amounts to projecting the data to a lower dimensional space: given an input vector $\mathbf{x}$, the mapping consists of computing the projections of $\mathbf{x}$ along the $\mathbf{e}_a$, $a = 1, \ldots, d'$, thereby constructing the components of the projected d'-dimensional feature vectors. Finally, since the d directions are orthogonal, they also provide a complete basis. Thus if one uses all d directions, no information is lost; and as we'll see below, given that one wants to project to a $d' < d$ dimensional space, if one uses the d' principal directions, then the mean squared error introduced by representing the data by their projections along these directions is minimized.

2.1.3 PCA Decorrelates the Data

Now suppose we've performed PCA on our samples, and instead of using it to construct low dimensional features, we simply use the full set of orthonormal eigenvectors as a choice of basis. In the old basis, a given input vector $\mathbf{x}$ is expanded as $\mathbf{x} = \sum_{a=1}^{d} x_a \mathbf{u}_a$ for some orthonormal set $\{\mathbf{u}_a\}$, and in the new basis, the same vector is expanded as $\mathbf{x} = \sum_{b=1}^{d} \tilde{x}_b \mathbf{e}_b$, so $\tilde{x}_a \equiv \mathbf{x} \cdot \mathbf{e}_a = \mathbf{e}_a \cdot \sum_b x_b \mathbf{u}_b$. The mean $\boldsymbol{\mu} \equiv \frac{1}{m}\sum_i \mathbf{x}_i$ has components $\tilde{\mu}_a = \boldsymbol{\mu} \cdot \mathbf{e}_a$ in the new basis. The sample covariance matrix depends on the choice of basis: if C is the covariance matrix in the old basis, then the corresponding covariance matrix in the new basis is $\tilde{C}_{ab} \equiv \frac{1}{m}\sum_i (\tilde{x}_{ia} - \tilde{\mu}_a)(\tilde{x}_{ib} - \tilde{\mu}_b) = \frac{1}{m}\sum_i \{\mathbf{e}_a \cdot (\sum_p x_{ip}\mathbf{u}_p - \boldsymbol{\mu})\}\{(\sum_q x_{iq}\mathbf{u}_q - \boldsymbol{\mu}) \cdot \mathbf{e}_b\} = \mathbf{e}'_a C \mathbf{e}_b = \lambda_b \delta_{ab}$. Hence in the new basis the covariance matrix is diagonal and the samples are uncorrelated. It's worth emphasizing two points: first, although the covariance matrix can be viewed as a geometric object in that it transforms as a tensor (since it is a summed outer product of vectors, which themselves have a meaning independent of coordinate system), nevertheless, the notion of correlation is basis-dependent (data can be correlated in one basis and uncorrelated in another). Second, no assumptions regarding the distribution of X has been made here.

2.1.4 PCA: Reconstruction with Minimum Squared Error

The basis provided by the eigenvectors of the covariance matrix is also optimal for dimension reduction in the following sense. Again consider some arbitrary orthonormal basis $\{\mathbf{u}_a,\ a = 1, \ldots, d\}$, and take the first d' of these to perform the dimension reduction: $\tilde{\mathbf{x}} \equiv \sum_{a=1}^{d'} (\mathbf{x} \cdot \mathbf{u}_a)\mathbf{u}_a$. The chosen $\mathbf{u}_a$ form a basis for $\mathcal{R}^{d'}$, so we may take the components of the dimensionally reduced vectors to be $\mathbf{x} \cdot \mathbf{u}_a$, $a = 1, \ldots, d'$ (although here we leave $\tilde{\mathbf{x}}$ with dimension d). Define the reconstruction error summed over the dataset as $\sum_{i=1}^{m} \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|^2$. Again assuming that the eigenvectors $\{\mathbf{e}_a\}$ of the covariance matrix are indexed in order of non-increasing eigenvalues, then choosing those eigenvectors as basis vectors will give minimal reconstruction error, as we will show. If the data is not centered, then the mean should be subtracted first, the dimension reduction performed, and the mean then added back[8]; thus in this case, the dimensionally reduced data will still lie in the subspace $\mathcal{R}^{d'}$, but that subspace will be offset from the origin by the mean. Bearing this caveat in mind, to prove the claim we can assume that the data is centered. Expanding $\mathbf{u}_a \equiv \sum_{p=1}^{d} \beta_{ap} \mathbf{e}_p$, we have

$$\frac{1}{m}\sum_i \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|^2 = \frac{1}{m}\sum_i \|\mathbf{x}_i\|^2 - \frac{1}{m}\sum_{a=1}^{d'}\sum_i (\mathbf{x}_i \cdot \mathbf{u}_a)^2 \tag{2.4}$$

with orthogonality constraints $\sum_{p=1}^{d} \beta_{ap}\beta_{bp} = \delta_{ab}$. The second term on the right is

$$-\sum_{a=1}^{d'} \mathbf{u}_a^T C \mathbf{u}_a = -\sum_{a=1}^{d'} (\sum_{p=1}^{d} \beta_{ap}\mathbf{e}_p^T) C (\sum_{q=1}^{d} \beta_{aq}\mathbf{e}_q) = -\sum_{a=1}^{d'}\sum_{p=1}^{d} \lambda_p \beta_{ap}^2 \tag{2.5}$$

Introducing Lagrange multipliers ω_{ab} to enforce the orthogonality constraints [Burges, 2004], in order to minimize the reconstruction error we must maximize

$$F = \sum_{a=1}^{d'}\sum_{p=1}^{d} \lambda_p \beta_{ap}^2 - \sum_{a,b=1}^{d'} \omega_{ab} \left(\sum_{p=1}^{d} \beta_{ap}\beta_{bp} - \delta_{ab} \right) \tag{2.6}$$

[8] The principal eigenvectors are not necessarily the directions that give minimal reconstruction error if the data is not centered: imagine data whose mean is both orthogonal to the principal eigenvector and far from the origin. The single direction that gives minimal reconstruction error will be close to the mean.

Choosing[9] $\omega_{ab} \equiv \omega_a \delta_{ab}$ and taking derivatives with respect to β_{cq} gives $\lambda_q \beta_{cq} = \omega_c \beta_{cq}$. Both this and the constraints can be satisfied by choosing $\omega_a = \lambda_a$ and $\beta_{ap} = \delta_{ap}$ for $p \leq d'$, $\beta_{ap} = 0$ otherwise. The objective function then simply becomes $\sum_{p=1}^{d'} \lambda_p$, which is maximized by choosing the first d' largest λ_p. Note that this also amounts to a proof that, for projections that give minimal reconstruction error, the 'greedy' approach to PCA dimension reduction - solve for a single optimal direction (which gives the principal eigenvector as first basis vector), then project your data into the subspace orthogonal to that, then repeat - also results in the global optimal solution, found by solving for all directions at once. The same observation applies to finding projections that maximally reduce the residual variance. Again, note that this argument is distribution independent.

2.1.5 PCA Maximizes Mutual Information on Gaussian Data

Now consider some proposed set of projections $W \in M_{d'd}$, where the rows of W are orthonormal, so that the projected data is $\mathbf{y} \equiv W\mathbf{x}$, $\mathbf{y} \in \mathcal{R}^{d'}$, $\mathbf{x} \in \mathcal{R}^{d}$, $d' \leq d$. Suppose that $X \sim \mathcal{N}(0, C)$. Then since the $\mathbf{y}$'s are linear combinations of the $\mathbf{x}$'s, they are also normally distributed, with zero mean and sample covariance $C_y \equiv (1/m)\sum_i^m \mathbf{y}_i \mathbf{y}_i' = (1/m)W(\sum_i^m \mathbf{x}_i \mathbf{x}_i')W' = WCW'$. It's interesting to ask how W can be chosen so that the mutual information between the distribution of X and that of Y is maximized [Baldi and Hornik, 1995, Diamantaras and Kung, 1996]. Since the mapping W is deterministic, the conditional entropy $H(Y|X)$ vanishes, and the mutual information is just $I(X, Y) = H(Y) - H(Y|X) = H(Y)$. Using a small, fixed bin size, we can approximate this by the differential entropy,

$$H(Y) = -\int p(\mathbf{y}) \log_2 p(\mathbf{y}) d\mathbf{y} = \frac{1}{2}\log_2(e(2\pi)^{d'}) + \frac{1}{2}\log_2 \det(C_y) \tag{2.7}$$

This is maximized by maximizing $\det(C_y) = \det(WCW')$ over choice of W, subject to the constraint that the rows of W are orthonor-

[9] Recall that Lagrange multipliers can be chosen in any way that results in a solution satisfying the constraints.

mal. The general solution to this is $W = UE$, where U is an arbitrary d' by d' orthogonal matrix, and where the rows of $E \in M_{d'd}$ are formed from the first d' principal eigenvectors of C, and at the solution, $\det(C_y)$ is just the product of the first d' principal eigenvalues. Clearly, the choice of U does not affect the entropy, since $\det(UECE'U') = \det(U)\det(ECE')\det(U') = \det(ECE')$. In the special case where $d' = 1$, so that E consists of a single, unit length vector $\mathbf{e}$, we have $\det(ECE') = \mathbf{e}'C\mathbf{e}$, which is maximized by choosing $\mathbf{e}$ to be the principal eigenvector of C, as shown above. (The other extreme case, where $d' = d$, is easy too, since then $\det(ECE') = \det(C)$ and E can be any orthogonal matrix). We refer the reader to [Wilks, 1962] for a proof for the general case $1 < d' < d$.

2.2 Probabilistic PCA (PPCA)

Suppose you've applied PCA to obtain low dimensional feature vectors for your data, but that you have also somehow found a partition of the data such that the PCA projections you obtain on each subset are quite different from those obtained on the other subsets. It would be tempting to perform PCA on each subset and use the relevant projections on new data, but how do you determine what is 'relevant', and how in general would you even find such subsets? These problems could be addressed if we could learn a mixture of generative models for the data, where each model corresponded to its own PCA decomposition. [Tipping and Bishop, 1999A, Tipping and Bishop, 1999B] proposed such a model - "Probabilistic PCA" - building on earlier work linking PCA decomposition to factor analysis. The advantages of a probabilistic model are numerous: for example, the weight that each mixture component gives to the posterior probability of a given data point can be computed, solving the 'relevance' problem stated above. In this section we briefly review PPCA.

The approach is in fact a form of factor analysis, which itself is a classical dimension reduction technique. Factor analysis first appeared in the behavioral sciences community over a century ago, when Spearman hypothesised that intelligence could be reduced to a single underlying factor [Spearman, 1904]. If, given an n by n correlation matrix between variables $X_i \in \mathcal{R}$, $i = 1, \cdots, n$, there is a single variable g such that the conditional correlation between X_i and X_j vanishes for $i \neq j$ given the value of g, then g is the underlying 'factor' and the off-diagonal elements of the correlation matrix can be written as the corresponding off-diagonal elements of $\mathbf{z}\mathbf{z}'$ for some $\mathbf{z} \in \mathcal{R}^n$ [Darlington, 1997]. Modern factor analysis usually considers a model where the underlying factors $X \in \mathcal{R}^{d'}$ are Gaussian, and where a Gaussian noise term $\boldsymbol{\epsilon} \in \mathcal{R}^d$ is added:

$$\begin{aligned} Y &= WX + \boldsymbol{\mu} + \boldsymbol{\epsilon} \qquad (2.8) \\ X &\sim \mathcal{N}(0, \mathbf{1}) \\ \boldsymbol{\epsilon} &\sim \mathcal{N}(0, \Psi) \end{aligned}$$

Here $Y \in \mathcal{R}^d$ are the observations, the parameters of the model are $W \in M_{dd'}$ $(d' \leq d)$, Ψ and $\boldsymbol{\mu}$, and Ψ is assumed to be diagonal. By construction, Y has mean $\boldsymbol{\mu}$ and 'model covariance' $WW' + \Psi$. For this model, given X, the vectors $Y - \boldsymbol{\mu}$ become uncorrelated, and ϵ_i captures the variance that is unique to Y_i. Since X and $\boldsymbol{\epsilon}$ are Gaussian distributed, so is Y, and so the maximum likelihood estimate of $\boldsymbol{\mu}$ is just the empirical expectation of the $\mathbf{y}$'s. However, in general, W and Ψ must be estimated iteratively, using for example EM. There is an instructive exception to this [Basilevsky, 1994]. Suppose that $\Psi = \sigma^2\mathbf{1}$, so that the $d - d'$ smallest eigenvalues of the model covariance are the same and are equal to σ^2. Suppose also that S, the sample covariance of the $\mathbf{y}$'s, is equal to the model covariance; we can then read off d' as the multiplicity of the smallest eigenvalue σ^2 of S. Let $\mathbf{e}^{(j)}$ be the j'th orthonormal eigenvector of S with eigenvalue λ_j. Then it is straightforward to check that $W_{ij} = \sqrt{(\lambda_j - \sigma^2)}\mathbf{e}_i^{(j)}$, $i = 1, \cdots, d$, $j = 1, \cdots, d'$ satisfies $WW' + \Psi = S$ if the $\mathbf{e}^{(j)}$ are in principal order. The model thus arrives at the PCA directions, but in a probabilistic way. *Probabilistic* PCA (PPCA) assumes a model of the form (2.8) with $\Psi = \sigma^2\mathbf{1}$, but it drops the above assumption that the model and sample covariances are equal (which in turn means that σ^2 must now be estimated). The resulting maximum likelihood estimates of W and σ^2 can be written in closed form, as [Tipping and Bishop, 1999A]

$$W_{ML} = U(\Lambda - \sigma^2\mathbf{1})R \qquad (2.9)$$

$$\sigma^2_{ML} = \frac{1}{d-d'} \sum_{i=d'+1}^{d} \lambda_i \qquad (2.10)$$

where $U \in M_{dd'}$ is the matrix of the d' principal column eigenvectors of S, Λ is the corresponding diagonal matrix of principal eigenvalues, and $R \in M_{d'}$ is an arbitrary orthogonal matrix. Thus σ^2 captures the variance lost in the discarded projections and the PCA directions appear in the maximum likelihood estimate of W (and in fact re-appear in the expression for the expectation of X given Y, in the limit $\sigma \to 0$, in which case the components of X become the PCA projections of Y). This closed form result is rather striking in view of the fact that for general factor analysis (for example, for diagonal but non-isotropic Ψ) we must resort to an iterative algorithm. The probabilistic formulation makes PCA amenable to a rich variety of probabilistic methods: for example, PPCA allows one to perform PCA when some of the data has missing components; and d' (which so far we've assumed known) can itself be estimated using Bayesian arguments [Bishop, 1999]. Returning to the problem posed at the beginning of this Section, a mixture of PPCA models, each with weight $\pi_i \geq 0$, $\sum_i \pi_i = 1$, can be computed for the data using maximum likelihood and EM, thus giving a principled approach to combining several local PCA models [Tipping and Bishop, 1999B].

2.3 Kernel PCA

PCA is a linear method, in the sense that the reduced dimension representation is generated by linear projections (although the eigenvectors and eigenvalues depend non-linearly on the data), and this can severely limit the usefulness of the approach. Several versions of nonlinear PCA have been proposed (see e.g. [Diamantaras and Kung, 1996]) in the hope of overcoming this problem. In this section we describe one such algorithm called kernel PCA [Schölkopf et al., 1998]. Kernel PCA relies on the "kernel trick", the essence of which rests on the following observation: suppose you have an algorithm (for example, k'th nearest neighbour) which depends only on dot products of the data. Consider using the same algorithm on transformed data: $\mathbf{x} \to \mathbf{\Phi}(\mathbf{x}) \in \mathcal{F}$, where $\mathcal{F}$ is a (possibly infinite dimensional) vector space, which we will call feature space. Operating in $\mathcal{F}$, your algorithm depends only on the dot products $\mathbf{\Phi}(\mathbf{x}_i) \cdot \mathbf{\Phi}(\mathbf{x}_j)$. Now suppose there exists a (symmetric) 'kernel' function $k(\mathbf{x}_i, \mathbf{x}_j)$ such that for all $\mathbf{x}_i,\ \mathbf{x}_j \in \mathcal{R}^d$, $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{\Phi}(\mathbf{x}_i) \cdot \mathbf{\Phi}(\mathbf{x}_j)$. Then since your algorithm depends only on these dot products, you never have to compute $\mathbf{\Phi}(\mathbf{x})$ explicitly; you can always just substitute the kernel form. In fact this 'trick' is very general, and since it is widely used, we summarize it briefly here. Consider a Hilbert space $\mathcal{H}$ (a complete vector space for which an inner product is defined). We will take $\mathcal{H}$ to be a space whose elements are real valued functions defined over $\mathcal{R}^d$, for concreteness. Consider the set of linear evaluation functionals $\mathcal{I}_{\mathbf{x}} : f \in \mathcal{H} \to f(\mathbf{x}) \in \mathcal{R}$, indexed by $\mathbf{x} \in \mathcal{R}^d$. If every such linear functional is continuous, then there is a special function k_x associated with $\mathcal{H}$, also indexed by $\mathbf{x}$, and called a reproducing kernel, for which $\langle f, k_x \rangle = f(x)$. Such Hilbert spaces are called Reproducing Kernel Hilbert Spaces (RKHSs) and this particular relation is called the reproducing property. In particular, the function $k_{\mathbf{x}_1}$ evaluated at some other point $\mathbf{x}_2$ is defined as $k(\mathbf{x}_1, \mathbf{x}_2) \equiv k_{\mathbf{x}_1}(\mathbf{x}_2)$, and using the reproducing property on $k_{\mathbf{x}}$ itself yields $\langle k_{\mathbf{x}_1}, k_{\mathbf{x}_2} \rangle = k(\mathbf{x}_1, \mathbf{x}_2)$. From this follow the usual properties we associate with kernels - they are symmetric in their arguments, and are positive definite functions. Mapping the notation back to our informal description above, $\mathcal{F}$ is $\mathcal{H}$, and $\Phi(\mathbf{x})$ is k_x. RKHS's were first introduced as a method to work implicitly in high dimensional spaces (in which classifiers are linearly separating hyperplanes), by [Aizerman et al., 1964] in their theory of potential functions (although their formalism was not cast in terms of Hilbert spaces and kernels, the potential functions they introduced are kernels in RKHSs); RKHSs gained further traction in the work of [Kimeldorf and Wahba, 1971], who introduced the “Representer Theorem”, which shows that under general conditions, the solution to a general regularized optimization problem in an RKHS can be written as an expansion over functions $k_{\mathbf{x}_i}$, where the $\mathbf{x}_i$ are training samples; and RKHSs appeared on the machine learning scene in [Boser et al., 1992], where they were first applied to support vector machines, to obtain classifiers that, although linear in the RKHS, are nonlinear when viewed as functions over the sample space.

Kernel PCA applies the kernel trick to create a nonlinear version of PCA in sample space by performing ordinary PCA in $\mathcal{F}$. It's striking that, since projections are being performed in a space whose dimension can be much larger than d, the number of useful such projections can actually exceed d. It is not immediately obvious that PCA is eligible for the kernel trick, since in PCA the data appears in expectations over products of individual components of vectors, not over inner products between the vectors. However [Schölkopf et al., 1998] show how the problem can indeed be cast entirely in terms of inner products. They make two key observations: first, that the eigenvectors of the covariance matrix in $\mathcal{F}$ lie in the span of the (centered) mapped data, and second, that therefore no information in the eigenvalue equation is lost if the equation is replaced by m equations, formed by taking the inner product of each side of the eigenvalue equation with each (centered) mapped data point. Let's see how this works. The covariance matrix of the mapped data in feature space is

$$C \equiv \frac{1}{m}\sum_{i=1}^{m}(\mathbf{\Phi}_i - \boldsymbol{\mu})(\mathbf{\Phi}_i - \boldsymbol{\mu})^T \tag{2.11}$$

where $\mathbf{\Phi}_i \equiv \mathbf{\Phi}(\mathbf{x}_i)$ and $\boldsymbol{\mu} \equiv \frac{1}{m}\sum_i \mathbf{\Phi}_i$. We are looking for eigenvector solutions $\mathbf{v}$ of

$$C\mathbf{v} = \lambda\mathbf{v} \tag{2.12}$$

Since this can be written $\frac{1}{m}\sum_{i=1}^{m}(\mathbf{\Phi}_i - \boldsymbol{\mu})[(\mathbf{\Phi}_i - \boldsymbol{\mu})\cdot\mathbf{v}] = \lambda\mathbf{v}$, the eigenvectors $\mathbf{v}$ lie in the span of the $\mathbf{\Phi}_i - \boldsymbol{\mu}$'s, or

$$\mathbf{v} = \sum_i \alpha_i(\mathbf{\Phi}_i - \boldsymbol{\mu}) \tag{2.13}$$

for some α_i. We will denote the vector whose ith component is α_i by $\boldsymbol{\alpha} \in \mathcal{R}^m$. Since (both sides of) Eq. (2.12) lie in the span of the $\mathbf{\Phi}_i - \boldsymbol{\mu}$, we can replace it with the m equations

$$(\mathbf{\Phi}_i - \boldsymbol{\mu})^T C\mathbf{v} = \lambda(\mathbf{\Phi}_i - \boldsymbol{\mu})^T\mathbf{v} \tag{2.14}$$

We can easily compute the kernel matrix K_{ij}, the matrix of inner products[10] in $\mathcal{F}$: $K_{ij} \equiv \langle\mathbf{\Phi}_i, \mathbf{\Phi}_j\rangle = k(\mathbf{x}_i, \mathbf{x}_j),\ i, j = 1, \ldots, m$. However, what

[10] A matrix of inner products is called a Gram matrix. Any Gram matrix G is necessarily

we need is the centered kernel matrix, $K^C_{ij} \equiv \langle(\boldsymbol{\Phi}_i - \boldsymbol{\mu}), (\boldsymbol{\Phi}_j - \boldsymbol{\mu})\rangle$. Note that we do not need to compute $\boldsymbol{\mu}$ explicitly since any m by m inner product matrix can be centered by left- and right- multiplying by the projection matrix $P \equiv \mathbf{1} - \frac{1}{m}\mathbf{e}\mathbf{e}'$, where $\mathbf{1}$ is the unit matrix in M_m and where $\mathbf{e}$ is the m-vector of all ones (see Section 3.2 for further discussion of centering). Thus $K^C = PKP$ and Eq. (2.14) becomes

$$K^C K^C \boldsymbol{\alpha} = \nu K^C \boldsymbol{\alpha} \tag{2.15}$$

where $\nu \equiv m\lambda$. Now clearly any solution of

$$K^C \boldsymbol{\alpha} = \nu \boldsymbol{\alpha} \tag{2.16}$$

is also a solution of (2.15). It's straightforward to show that any solution of (2.15) can be written as a solution $\boldsymbol{\alpha}$ to (2.16) plus a vector $\boldsymbol{\beta}$ which is orthogonal to $\boldsymbol{\alpha}$ (and which satisfies $\sum_i \beta_i(\boldsymbol{\Phi}_i - \boldsymbol{\mu}) = 0$), and which therefore does not contribute to (2.13); therefore we need only consider Eq. (2.16). Finally, to use the eigenvectors $\mathbf{v}$ to compute principal components in $\mathcal{F}$, we need $\mathbf{v}$ to have unit length, that is, $\mathbf{v}\cdot\mathbf{v} = 1 = \nu\boldsymbol{\alpha}\cdot\boldsymbol{\alpha}$ (using (2.13) and (2.16)), so the $\boldsymbol{\alpha}$ must be normalized to have length $1/\sqrt{\nu}$.

The recipe for extracting the i'th principal component in $\mathcal{F}$ using kernel PCA is therefore:

(1) Compute the i'th principal eigenvector of K^C, with eigenvalue ν.

(2) Normalize the corresponding eigenvector, $\boldsymbol{\alpha}$, to have length $1/\sqrt{\nu}$.

(3) For a training point $\mathbf{x}_k$, the principal component is then just

$$(\boldsymbol{\Phi}(\mathbf{x}_k) - \boldsymbol{\mu}) \cdot \mathbf{v} = \nu\alpha_k \tag{2.17}$$

(4) For a general test point $\mathbf{x}$, the principal component is

$$\begin{aligned}(\boldsymbol{\Phi}(\mathbf{x}) - \boldsymbol{\mu}) \cdot \mathbf{v} &= \sum_i \alpha_i k(\mathbf{x}, \mathbf{x}_i) - \frac{1}{m}\sum_{i,j} \alpha_i k(\mathbf{x}, \mathbf{x}_j) \\ &- \frac{1}{m}\sum_{i,j} \alpha_i k(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{m^2}\sum_{i,j,n} \alpha_i k(\mathbf{x}_j, \mathbf{x}_n)\end{aligned}$$

positive semidefinite, as is easily seen in this case from $\mathbf{z}'K\mathbf{z} = \sum_{ij} z_i z_j \langle\Phi_i, \Phi_j\rangle = \|\sum_i z_i \Phi_i\|^2$.

where the last two terms can be dropped since they don't depend on $\mathbf{x}$.

Kernel PCA may be viewed as a way of putting more effort into the up-front computation of features, rather than putting the onus on the classifier or regression algorithm. Kernel PCA followed by a linear SVM on a pattern recognition problem has been shown to give similar results to using a nonlinear SVM using the same kernel [Schölkopf et al., 1998]. It shares with other kernel methods the attractive property of mathematical tractability and of having a clear geometrical interpretation: for example, this has led to using kernel PCA for de-noising data, by finding that vector $\mathbf{z} \in \mathcal{R}^d$ such that the Euclidean distance between $\Phi(\mathbf{z})$ and the vector computed from the first few PCA components in $\mathcal{F}$ is minimized [Mika et al., 1999]. Classical PCA has the significant limitation that it depends only on first and second moments of the data, whereas kernel PCA does not (for example, a polynomial kernel $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + b)^p$ contains powers up to order $2p$, which is particularly useful for e.g. image classification, where one expects that products of several pixel values will be informative as to the class). Kernel PCA has the computational limitation of having to compute eigenvectors for square matrices of side m, but again this can be addressed, for example by using a subset of the training data, or by using the Nyström method for approximating the eigenvectors of a large Gram matrix (see below).

2.4 Canonical Correlation Analysis

Suppose we have two datasets $\mathbf{x}_{1i} \in S_1, \mathbf{x}_{2j} \in S_2$, where $S_1 \equiv R^{d_1}$, $S_2 \equiv R^{d_2}$, and $i = 1, ..., m_1$, $j = 1, ..., m_2$. Note that d_1 may not equal d_2 and that m_1 may not equal m_2. Canonical Correlation Analysis (CCA) [Hotelling, 1936] finds paired directions $\{\mathbf{w}_1, \mathbf{w}_2\}$, $\mathbf{w}_1 \in S_1$, $\mathbf{w}_2 \in S_2$ such that the projection of the first dataset along $\mathbf{w}_1$ is maximally correlated with the projection of the second dataset along $\mathbf{w}_2$. In addition, for $i \neq j$, the projections along the pairs $\{\mathbf{w}_{1i}, \mathbf{w}_{1j}\}$, $\{\mathbf{w}_{2i}, \mathbf{w}_{2j}\}$ and $\{\mathbf{w}_{1i}, \mathbf{w}_{2j}\}$ are all uncorrelated. Furthermore, the values of the $\mathbf{w} \cdot \mathbf{x}$'s themselves are invariant to invertible affine transformations of the data, which gives CCA a coordinate independent meaning, in contrast to ordinary correlation analysis. Hotelling gives the following example, taken from [Kelly, 1928]: 140 seventh-grade school children were tested for their ability in reading and arithmetic. Two measures of efficacy were used for reading (speed and 'power') and two for arithmetic (also called speed and 'power'). In this case CCA revealed that, according to this data, reading and arithmetic involve one and only one common mental factor, with a p-value of approximately 0.0001. Note that in cases like this, the datasets are themselves paired, so that $m_1 = m_2$.

CCA may be summarized as follows (in this section, we will reserve the subscripts $\{p, q\}$ to lie in $\{1, 2\}$, and we remind the reader that subscripts $\{i, j\}$ index vectors, and $\{a, b\}$ index vector components). We are given two random variables, $\mathbf{X}_1$, $\mathbf{X}_2$ with sample spaces $\Omega_1 \in R^{d_1}$ and $\Omega_2 \in R^{d_2}$. We assume for now that we are able to compute expectations of quadratic products of the $\mathbf{X}$'s. To keep the discussion uncluttered we also assume that both $\mathbf{X}_1$ and $\mathbf{X}_2$ have zero mean. Introduce random variables $U \equiv \mathbf{X}_1 \cdot \mathbf{w}_1$ and $V \equiv \mathbf{X}_2 \cdot \mathbf{w}_2$. We wish to find $\mathbf{w}_1 \in \mathcal{R}^{d_1}$, $\mathbf{w}_2 \in \mathcal{R}^{d_2}$, such that the correlation

$$\rho \equiv \frac{E[UV]}{\sqrt{E[U^2]E[V^2]}} = \frac{\mathbf{w}_1' C_{12} \mathbf{w}_2}{\sqrt{(\mathbf{w}_1' C_{11} \mathbf{w}_1)(\mathbf{w}_2' C_{22} \mathbf{w}_2)}} \equiv \frac{A_{12}}{\sqrt{A_{11} A_{22}}} \tag{2.18}$$

is maximized, where $C_{pq} \equiv E[\mathbf{X}_p \mathbf{X}_q']$ and we have introduces scalars $A_{pq} \equiv \mathbf{w}_p' C_{pq} \mathbf{w}_q$. Setting the derivative of ρ^2 with respect to w_{pa} equal to zero for $p = \{1, 2\}$ gives

$$C_{11}^{-1} C_{12} \mathbf{w}_2 = \frac{A_{12}}{A_{11}} \mathbf{w}_1 \tag{2.19}$$

$$C_{11}^{-1} C_{12} C_{22}^{-1} C_{21} \mathbf{w}_1 = \rho^2 \mathbf{w}_1 \tag{2.20}$$

(where we have assumed that the covariance matrices C_{11} and C_{22} are nonsingular; note that $A_{12} = A_{21}$ and that $C_{12} = C_{21}'$), and similarly for $\{1 \Leftrightarrow 2\}$.

The matrices left-multiplying the $\mathbf{w}$'s in Eqs. (2.19, 2.20) are not necessarily symmetric (note that C_{12} is not necessarily square). Since the eigenvalues of general square matrices need not be real, it would be comforting to check that solving Eqs. (2.19, 2.20) will always result in real, positive ρ^2. We can use Choleski decomposition to write $C_{pp} \equiv R_{pp} R_{pp}'$ where R_{pp} is lower triangular [Horn and Johnson, 1985]: then writing $\mathbf{z}_1 \equiv R'_{11}\mathbf{w}_1$, Eq. (2.20) becomes

$$R_{11}^{-1} C_{12} C_{22}^{-1} C_{21} R_{11}'^{-1} \mathbf{z}_1 = \rho^2 \mathbf{z}_1 \tag{2.21}$$

The left hand multiplicand is now a (symmetric) positive definite matrix, since for any vector $\mathbf{s} \in R^{d_1}$, we have

$$\mathbf{s}' R_{11}^{-1} C_{12} C_{22}^{-1} C_{21} R_{11}'^{-1} \mathbf{s} = \mathbf{t}'\mathbf{t} \tag{2.22}$$

where $\mathbf{t} \equiv R_{22}^{-1} C_{21} R_{11}^{-1}\mathbf{s}$, so ρ^2 is indeed real and positive.

While we're on the subject of sanity checks, it's conceivable that (2.20) is necessary but not sufficient, so we should also check that its eigenvalues always take the form $\rho^2 = A_{12}^2/(A_{11}A_{22})$. Eq. (2.20) and the $\{1 \Leftrightarrow 2\}$ version of Eq. (2.19) gives

$$\rho^2 A_{11} = \mathbf{w}_1' C_{12} C_{22}^{-1} C_{21} \mathbf{w}_1 = \mathbf{w}_1' C_{12} \frac{A_{12}}{A_{22}} \mathbf{w}_2 = \frac{A_{12}^2}{A_{22}} \tag{2.23}$$

2.4.1 CCA Decorrelates the Data

CCA shares with PCA the property that the projections decorrelate the data. For CCA, the data is decorrelated both within S_1 and S_2 and between S_1 and S_2, and the directions are conjugate with respect to the covariance matrices. To see this, consider the set of solutions $\mathbf{w}_{1i}$ and corresponding $\mathbf{w}_{2i}$. First note that from Eq. (2.20), for $\rho_i \neq \rho_j$,

$$\mathbf{w}_{1j}' C_{12} C_{22}^{-1} C_{21} \mathbf{w}_{1i} = \rho_i^2 \mathbf{w}_{1j}' C_{11} \mathbf{w}_{1i} = \rho_j^2 \mathbf{w}_{1i}' C_{11} \mathbf{w}_{1j} = 0 \tag{2.24}$$

Hence $\mathbf{w}_{1i}' C_{11} \mathbf{w}_{1j} = 0 = \mathbf{w}_{2i}' C_{22} \mathbf{w}_{2j}$. Similarly from (2.19), we have $\mathbf{w}_{2j}' C_{21} \mathbf{w}_{1i} = (A_{12}/A_{22}) \mathbf{w}_{2j}' C_{22} \mathbf{w}_{2i} = 0$, again for distinct eigenvalues. For repeated eigenvalues, the $\mathbf{w}$'s may again be chosen to be conjugate with respect to the covariance matrices. Thus in the new basis, the variables are uncorrelated:

$$E[U_i U_j'] = E[\mathbf{w}_{1i} \cdot \mathbf{X}_1 \mathbf{w}_{1j} \cdot \mathbf{X}_1] = \mathbf{w}_{1i}' C_{11} \mathbf{w}_{1j} = 0 \text{ for } i \neq j \tag{2.25}$$

and similarly $E[V_i V_j'] = E[U_i V_j'] = 0$ if $i \neq j$.

2.4.2 CCA is Invariant under Invertible Affine Transformations

What happens to the $\mathbf{w} \cdot \mathbf{x}$ projections if we translate, rotate or scale the data? Perhaps we should whiten the data in both S_1 and S_2 first. One of the strengths of CCA is that this is not necessary: the projected values are invariant under invertible affine transformations $\mathbf{x} \in R^d \rightarrow B\mathbf{x} + b, \; B \in M_d, \; b \in R^d$, provided the $\mathbf{w}$'s are appropriately transformed.

Invariance with respect to translations follows directly from the definition of ρ, since covariance matrices are functions of the centered data. We can check invariance under the invertible transformation $\bar{\mathbf{x}}_1 \equiv B\mathbf{x}_1$ as follows: in the new coordinate system, Eq. (2.20) becomes

$$\bar{C}_{11}^{-1}\bar{C}_{12}C_{22}^{-1}\bar{C}_{21}\bar{\mathbf{w}}_1 = \rho^2\bar{\mathbf{w}}_1 \tag{2.26}$$

where $\bar{C}_{11} = BC_{11}B'$, $\bar{C}_{12} = BC_{12}$ and $\bar{C}_{21} = C_{21}B'$, so that

$$C_{11}^{-1}C_{12}C_{22}^{-1}C_{21}B'\bar{\mathbf{w}} = \rho^2 B'\bar{\mathbf{w}} \tag{2.27}$$

Hence the eigenvalues ρ take the same values. Thus solving in the transformed coordinate system we see that we will find $\bar{\mathbf{w}}_1$ which are related to $\mathbf{w}_1$ by $\mathbf{w}_1 = B'\bar{\mathbf{w}}_1$, so that for any $\mathbf{x}_1 \in S_1$, $\mathbf{w}_1 \cdot \mathbf{x}_1 = (\bar{\mathbf{w}}_1 B) \cdot (B^{-1}\bar{\mathbf{x}}_1) = \bar{\mathbf{w}}_1 \cdot \bar{\mathbf{x}}_1$. Thus the projections themselves remain invariant, and hence the correlations between projections remain invariant. By simply swapping $\{1 \leftrightarrow 2\}$ in the above argument we see that in this sense, CCA is invariant under invertible affine transformations of both S_1 and S_2 independently.

Note that the above two properties - decorrelation and affine invariance - are not shared by ordinary correlation analysis: the matrix whose ab'th element is $E[x_{1a}x_{2b}]/\sqrt{E[x_{1a}^2]E[x_{2b}^2]}$ can take very different forms in different coordinate systems. For example, given a set of random variables that are distributed as the components of a multivariate Gaussian, one can choose an affine transformation to a new coordinate system in which the data are uncorrelated: correlation alone is a coordinate dependent concept.

2.4.3 CCA in Practice; Kernel CCA

So far we have assumed that expectations of products of the $\mathbf{X}$'s can be computed. This is usually a tall order, and in practice CCA is applied to paired datasets, $\mathbf{x}_{1i}, \; \mathbf{x}_{2j}, \; i, j = 1, ..., m$, where the empirical distribution (used to compute the above expectations) is simply $P(x_{pia}, x_{qjb}) = (1/m)\delta_{ij}$, giving covariance matrices (for zero mean data) $C_{pq} = (1/m)\sum_{i=1}^{m} \mathbf{x}_{pi}\mathbf{x}'_{qi}$. The underlying assumption here is that $\mathbf{x}_{1i}$ and $\mathbf{x}_{2i}$ are different views of the same object (for example, measurements of mathematical ability, and reading ability, for the i'th seventh-grader). While CCA was originally proposed as a kind of factor analysis for paired datasets, the projections can also be used as (heuristic) similarity measures: [Hardoon, Szedmak and Shawe-Taylor, 2004] consider the problem of content-based image retrieval from the web, where vectors in S_1 represent the image content and vectors in S_2 represent the text surrounding the image. At run time, the user enters some text, and an appropriate image is hopefully retrieved. The similarity of a piece of text and an image is defined as the cosine between the vector whose ith component is $\mathbf{x}_1 \cdot \mathbf{w}_{1i}$ and the vector whose ith component is $\mathbf{x}_2 \cdot \mathbf{w}_{2i}$; the lookup requires a scan over the image database for each incoming text query. [Hardoon, Szedmak and Shawe-Taylor, 2004] also show how CCA can be kernelized, which we summarize briefly here.

Kernel CCA follows kernel PCA in spirit. The data $\mathbf{x}_1 \in R^{d_1}$, $\mathbf{x}_2 \in R^{d_2}$ are mapped to feature spaces $\mathcal{F}_1$ and $\mathcal{F}_2$ by maps Φ_1, Φ_2 respectively (note that $\mathcal{F}_1$ and $\mathcal{F}_2$ may or may not be the same). Since the $\mathbf{w}_{1i} \in \mathcal{F}_1$, $\mathbf{w}_{2i} \in \mathcal{F}_2$ are used only to take projections, we can assume that they lie in the span of the data, so that there exist α_p such that

$$\mathbf{w}_p = \sum_{i=1}^{m} \alpha_{pi}\Phi_p(\mathbf{x}_{pi}) \tag{2.28}$$

where we have dropped the index enumerating the $\mathbf{w}$'s (and the corresponding index on the α's) for clarity. Thus, for a given solution, $\alpha_p \in R^m$. Since CCA depends only on inner products the Φ's are never explicitly needed:

$$\mathbf{w}_p \cdot \Phi_p(\mathbf{x}_{pj}) = \sum_{i=1}^{m} \alpha_{pi}\langle \Phi_p(\mathbf{x}_{pi}), \Phi_p(\mathbf{x}_{pj})\rangle = \sum_{i=1}^{m} \alpha_{pi} K_p(\mathbf{x}_{pi}, \mathbf{x}_{pj}) \tag{2.29}$$

Following the above analysis, but in the spaces $\mathcal{F}_p$, yields

$$\rho = \max_{\alpha_1,\alpha_2} \frac{\alpha'_1 K_1 K_2 \alpha_2}{\sqrt{\alpha'_1 K_1^2 \alpha_1 \alpha'_2 K_2^2 \alpha_2}} \tag{2.30}$$

where each K_p is a square matrix of side m. For any data and mapping Φ_p for which the K's are invertible, this can be solved analytically;

however the solutions have perfect correlation (or anti correlation): $\rho = \pm 1$. An example of such a choice of $\mathcal{F}$ is the space corresponding to radial basis function kernels. Such a mapping clearly gives too much 'wiggle room' to the data; we need to regularize. This can be achieved with the same regularization device used in partial least squares, by penalizing the norm of the $\mathbf{w}_p$ vectors. For large datasets, and additional problem must be addressed: matrices of side equal to the sample size must be inverted. This can be overcome using approximate techniques such as incomplete Choleski decomposition. We refer the reader to [Hardoon, Szedmak and Shawe-Taylor, 2004] for details.

2.5 Oriented PCA and Distortion Discriminant Analysis

Before leaving projective methods, we describe another extension of PCA, which has proven very effective at extracting robust features from audio [Burges et al., 2002, Burges et al., 2003]. We first describe the method of oriented PCA (OPCA) [Diamantaras and Kung, 1996]. Suppose we are given a set of 'signal' vectors $\mathbf{x}_i \in \mathcal{R}^d$, $i = 1, \ldots, m$, where each $\mathbf{x}_i$ represents an undistorted data point, and suppose that for each $\mathbf{x}_i$, we have a set of N distorted versions $\tilde{\mathbf{x}}_i^k,\ k = 1, \ldots, N$. Define the corresponding 'noise' difference vectors to be $\mathbf{z}_i^k \equiv \tilde{\mathbf{x}}_i^k - \mathbf{x}_i$. Roughly speaking, we wish to find linear projections which are as orthogonal as possible to the difference vectors, but along which the variance of the signal data is simultaneously maximized. Denote the unit vectors defining the desired projections by $\mathbf{n}_i,\ i = 1, \ldots, d',\ \mathbf{n}_i \in \mathcal{R}^d$, where d' will be chosen by the user. By analogy with PCA, we could construct a feature extractor $\mathbf{n}$ which minimizes the mean squared reconstruction error $\frac{1}{mN}\sum_{i,k}(\mathbf{x}_i - \hat{\mathbf{x}}_i^k)^2$, where $\hat{\mathbf{x}}_i^k \equiv (\tilde{\mathbf{x}}_i^k \cdot \mathbf{n})\mathbf{n}$. The $\mathbf{n}$ that solves this problem is that eigenvector of $R_1 - R_2$ with largest eigenvalue, where R_1, R_2 are the correlation matrices of the $\mathbf{x}_i$ and $\mathbf{z}_i$ respectively. However this feature extractor has the undesirable property that the direction $\mathbf{n}$ will change if the noise and signal vectors are globally scaled with two different scale factors. OPCA [Diamantaras and Kung, 1996] solves this problem. The first OPCA direction is defined as that direction $\mathbf{n}$ that maximizes the generalized Rayleigh quotient [Duda and Hart, 1973, Diamantaras and Kung, 1996] $q_0 = \frac{\mathbf{n}' C_1 \mathbf{n}}{\mathbf{n}' C_2 \mathbf{n}}$, where C_1 is the covariance matrix of the signal and C_2 that of the noise. For d' directions collected into a column matrix $\mathcal{N} \in M_{dd'}$, we instead maximize $\frac{\det(\mathcal{N}'C_1\mathcal{N})}{\det(\mathcal{N}'C_2\mathcal{N})}$. For Gaussian data, this amounts to maximizing the ratio of the volume of the ellipsoid containing the data, to the volume of the ellipsoid containing the noise, where the volume is that lying inside an ellipsoidal surface of constant probability density. We in fact use the correlation matrix of the noise rather than the covariance matrix, since we wish to penalize the mean noise signal as well as its variance (consider the extreme case of noise that has zero variance but nonzero mean). Explicitly, we take

$$C \equiv \frac{1}{m}\sum_i (\mathbf{x}_i - E[\mathbf{x}])(\mathbf{x}_i - E[\mathbf{x}])' \tag{2.31}$$

$$R \equiv \frac{1}{mN}\sum_{i,k} \mathbf{z}_i^k (\mathbf{z}_i^k)' \tag{2.32}$$

and maximize $q = \frac{\mathbf{n}'C\mathbf{n}}{\mathbf{n}'R\mathbf{n}}$, whose numerator is the variance of the projection of the signal data along the unit vector $\mathbf{n}$, and whose denominator is the projected mean squared "error" (the mean squared modulus of all noise vectors $\mathbf{z}_i^k$ projected along $\mathbf{n}$). We can find the directions $\mathbf{n}_j$ by setting $\nabla q = 0$, which gives the generalized eigenvalue problem $C\mathbf{n} = qR\mathbf{n}$; those solutions are also the solutions to the problem of maximizing $\frac{\det(\mathcal{N}'C\mathcal{N})}{\det(\mathcal{N}'R\mathcal{N})}$. If R is not of full rank, it must be regularized for the problem to be well-posed. It is straightforward to show that, for positive semidefinite C, R, the generalized eigenvalues are positive, and that scaling either the signal or the noise leaves the OPCA directions unchanged, although the eigenvalues will change. Furthermore the $\mathbf{n}_i$ are, or may be chosen to be, linearly independent, and although the $\mathbf{n}_i$ are not necessarily orthogonal, they are conjugate with respect to both matrices C and R, that is, $\mathbf{n}_i'C\mathbf{n}_j \propto \delta_{ij}$, $\mathbf{n}_i'R\mathbf{n}_j \propto \delta_{ij}$.

OPCA is similar mathematically to multiclass discriminant analysis (MDA) where the number of classes is equal to m [Duda and Hart, 1973], but there is a crucial difference: in MDA, there is no notion of a 'canonical' (or 'signal') sample for each class; the MDA within-class scatter for a given class is computed as the covariance matrix for that class. In OPCA, for each class, the mean vector used in MDA is replaced by the single canonical (zero noise) point, which can lie far from the sample mean. This is done in both the numerator, where each MDA class means is replaced by the corresponding signal point, and where the overall mean is replaced by the mean of the signal points; and in the denominator, where the sample covariance for a given class is replaced by sums of squares of differences between noise vectors for that class and the signal vector for that class. This amounts to leveraging additional, valuable information about the problem, and can lead to significantly improved results for problems where such data is available (such as the audio fingerprinting task, where a very clean version of each original clip can be obtained).

'Distortion discriminant analysis' [Burges et al., 2002, Burges et al., 2003] uses layers of OPCA projectors both to reduce dimensionality (a high priority for audio or video data) and to make the features more robust. The above features, computed by taking projections along the **n**'s, are first translated and normalized so that the signal data has zero mean and the noise data has unit variance. For the audio application, for example, the OPCA features are collected over several audio frames into new 'signal' vectors, the corresponding 'noise' vectors are measured, and the OPCA directions for the next layer found. This has the further advantage of allowing different types of distortion to be penalized at different layers, since each layer corresponds to a different time scale in the original data (for example, a distortion that results from comparing audio whose frames are shifted in time to features extracted from the original data - 'alignment noise' - can be penalized at larger time scales).

2.6 Sufficient Dimension Reduction

This review is aimed primarily at dimension reduction in the unsupervised setting, but OPCA could be considered as attaching a label to each signal point, and in this section we continue exploring this supervisory thread by considering techniques for dimension reduction where the data consists of predictor-response pairs $\{\mathbf{x}_i, y_i\},\ i = 1, \ldots, m$. There are several reasons one might want to do this: if the **x**'s appear in the underlying functional dependence only through a small number of projections, then those projections may be used to construct various plots to visualize the data; and smaller, more accurate models of the regression itself can be constructed if one knows that an entire subspace can be ignored.

We consider models of the form

$$y = f(\mathbf{a}_1'\mathbf{x}, \mathbf{a}_2'\mathbf{x}, ..., \mathbf{a}_k'\mathbf{x}, \epsilon), \quad \mathbf{a}_i, \mathbf{x} \in \mathcal{R}^d,\ \epsilon \in \mathcal{R} \tag{2.33}$$

where the ϵ's model the noise and are assumed independent of X. The presence of the y's can drastically change the picture: for example, X could be distributed in such a way that no useful dimension reduction of the X alone is possible, whereas Y might depend only on a single component of X. Let $\mathbf{a}$ denote that matrix whose columns are the $\mathbf{a}_i$. Then the above list of dot products may be written as $\mathbf{a}^T\mathbf{x}$ and this defines a subspace spanned by the columns of $\mathbf{a}^T$, which we will denote by $S_\mathbf{a}$. An alternative way of writing (2.33), which emphasizes its underlying statistical independence assumption, is

$$Y \perp\!\!\!\perp X \mid \mathbf{a}^T X \tag{2.34}$$

Here we will follow the notation of [Cook, 1998], who defines a *minimum dimension-reduction subspace* (minimum DRS) as a space $S_\mathbf{a}$ satisfying Eq. (2.34) for which k is minimal. A *central subspace* is defined as the intersection of all DRS's, if that intersection is itself a DRS. If so, then the central subspace is the unique minimum DRS. However a central subspace may not in fact be a DRS (it's easy to construct an example for which one minimum DRS is spanned by x_1 and another by x_2 [Cook, 1998], so that the intersection is null). The goal of Sufficient Dimension Reduction[11] is to estimate the central subspace, when it exists (or, here, to estimate a minimum DRS when it doesn't). Let's start by describing one of the earliest approaches to SDR.

2.6.1 Sliced Inverse Regression

Sliced Inverse Regression (SIR) was introduced in a seminal paper by [Li, 1991a]. Since this paper sparked a fruitful line of research, we will

[11] The phrase *Sufficient Dimension Reduction* was introduced to the statistics community by [Cook and Lee, 1999]. The phrase *Sufficient Dimensionality Reduction* was introduced to the machine learning community by [Globerson and Tishby, 2003]. The approaches are quite different; we briefly summarize the latter below.

examine the basic ideas in detail here. Normal (forward) regression estimates $E[Y|\mathbf{x}]^{12}$. Inverse regression instead estimates $E[X|y]$, which is a much easier problem since it amounts to solving d one dimensional regression problems: $E[X|y]_a = \int x_a p(\mathbf{x}|y) d\mathbf{x} = \int x_a p(x_a|y) dx_a = E[X_a|y]$. It is a remarkable fact that a minimum DRS for the above problem (Eq. (2.33)) can be estimated when the marginal $p(X)$ is elliptic[13], and assuming that the $\mathbf{x}_i$ are IID. This can be done despite the fact that the problem as stated is ill-posed (i.e. by redefining f, the α's can be redefined) and despite the fact that we know nothing about f or ϵ directly. As y varies, $E[X|y]$ will trace a curve in $\mathcal{R}^d$. Noting that, given the form (2.33), for fixed ϵ, a small change $\mathbf{x} \to \mathbf{x} + \delta\mathbf{x}$ in the subspace orthogonal to $S_{\mathbf{a}}$ leaves y unchanged, one might hope to find conditions under which $E[X|y]$ can be shown to lie in $S_{\mathbf{a}}$. [Li, 1991a] gives us one such sufficient condition. Given this, the SIR algorithm consists of grouping the measured $\mathbf{x}_i$ by their corresponding values of y (binned if necessary), computing the mean for each group, and performing a weighted PCA on the resulting set of vectors in order to estimate $S_{\mathbf{a}}$.

Theorem 2.1. Given Eq. (2.33), further assume that $E[X|\mathbf{a}_1'\mathbf{x}, \mathbf{a}_2'\mathbf{x}, \ldots, \mathbf{a}_k'\mathbf{x}]$ lies in the subspace spanned by $\Sigma_{XX}\mathbf{a}_i$, where Σ_{XX} is the covariance matrix of X. Then the centered inverse regression curve $E[X|y] - E[X]$ lies in that subspace.

Proof. Here we will sacrifice brevity and a little generality in the cause of gaining further insight: we will assume that $p(X)$ is elliptic, and first show that this leads to the condition in the proof. We will denote the hyperplane defined by $\cap_i^k\{\mathbf{x} : \mathbf{a}_i'\mathbf{x} = \alpha_i\}$, simply by $\{\mathbf{a}_i'\mathbf{x} = \alpha_i\}$. First note that for *any* density $p(X)$, $E[X|\{\mathbf{a}_i'\mathbf{x} = \alpha_i\}]$ must itself lie on the hyperplane $\{\mathbf{a}_i'\mathbf{x} = \alpha_i\}$, since

$$\mathbf{a}_j' E[X|\{\mathbf{a}_i'\mathbf{x} = \alpha_i\}] = E[\mathbf{a}_j' X|\{\mathbf{a}_i'\mathbf{x} = \alpha_i\}] = E[\alpha_j|\{\mathbf{a}_i'\mathbf{x} = \alpha_i\}] = \alpha_j \tag{2.35}$$

[12] We restrict the disussion here to one dimensional y.

[13] An elliptic density is one for which the contours of constant density are ellipsoids, such as the Gaussian.

However this is not quite what we need: while this does mean that the expectation lies in a subspace, that subspace will vary as the α_i vary. We are interested in a stronger characterization of a subspace that depends on the $\mathbf{a}_i$ only.

Let's change coordinates to a basis in which the density is spherical[14], $\mathbf{z} = \Sigma_{XX}^{-1/2}\mathbf{x}$. Introducing $\mathbf{b}_i = \Sigma_{XX}^{1/2}\mathbf{a}_i$, in this coordinate system the constraints $\mathbf{a}_i'\mathbf{x} = \alpha_i$ become $\mathbf{a}_i'\Sigma_{XX}^{1/2}\mathbf{z} \equiv \mathbf{b}_i'\mathbf{z} = \alpha_i$. Consider the quantity

$$E[Z|\{\mathbf{b}_i'\mathbf{z} = \alpha_i\}] \tag{2.36}$$

The $\mathbf{b}_i$ need not be orthogonal: however we can always introduce an orthonormal set $\mathbf{u}_i$ such that for some β_i, the hyperplane

$$H \equiv \{\mathbf{a}_i'\mathbf{x} = \alpha_i\} = \{\mathbf{b}_i'\mathbf{z} = \alpha_i\} = \{\mathbf{u}_i'\mathbf{z} = \beta_i\} \tag{2.37}$$

Since $p(Z)$ is spherical and is centered at the origin, the induced density on H will also be spherical, and will be centered on the point of closest approach of H to the origin. Now points on H can be written as $\mathbf{z} = \sum_{i=1}^k \mathbf{u}_i\beta_i + \mathbf{u}_\perp$, where $\mathbf{u}_\perp'\mathbf{u}_i = 0$: the $\mathbf{u}_\perp$ lie in H. The nearest point on H to the origin is therefore $\sum_{i=1}^k \mathbf{u}_i\beta_i$, since $\mathbf{u}_\perp = 0$ minimizes $\|\sum_{i=1}^k \mathbf{u}_i\beta_i + \mathbf{u}_\perp\|_2$, and so transforming back to the $\mathbf{b}$'s, there must exist scalars γ_i such that

$$E[Z|\{\mathbf{u}_i'\mathbf{z} = \beta_i\}] = \sum_{i=1}^k \mathbf{u}_i\beta_i = \sum_{i=1}^k \mathbf{b}_i\gamma_i = \sum_{i=1}^k \Sigma_{XX}^{1/2}\mathbf{a}_i\gamma_i \tag{2.38}$$

Applying $\Sigma_{XX}^{1/2}$ to both sides gives the result:

$$E[X|\{\mathbf{a}_i'\mathbf{x} = \alpha_i\}] = \sum_{i=1}^k \Sigma_{XX}\mathbf{a}_i\gamma_i \tag{2.39}$$

so $E[X|\{\mathbf{a}_i'\mathbf{x} = \alpha_i\}]$ is spanned by the $\Sigma_{XX}\mathbf{a}_i$.

This geometric argument has brought us quite close to the desired result: we wish to show a similar result for

$$E[X|Y] = E[X|f(\mathbf{a}_1'\mathbf{x}, \mathbf{a}_2'\mathbf{x}, ..., \mathbf{a}_k'\mathbf{x}, \epsilon)] \tag{2.40}$$

[14] We assume that Σ_{XX} has been regularized if necessary so that Σ_{XX}^{-1} (and the density $p(X)$ itself) exists.

By the tower property [Ross and Peköz, 2007], and by conditional independence, $E[X|Y] = E[E[X|\{\mathbf{a}_i X\}, Y]|Y] = E[E[X|\{\mathbf{a}_i X\}]|Y]$, so by linearity of expectation, for any $\mathbf{s}$ in the space orthogonal to the $\Sigma_{XX}\mathbf{a}_i$, we have that $\mathbf{s}'E[X|Y] = E[E[\mathbf{s}'X|\{\mathbf{a}_i X\}]|Y] = 0$. □

We end this section with a brief tour of some related work. [Cook and Weisberg, 1991] point out that the above condition in fact implies that $p(\mathbf{x})$ is elliptical, although [Li, 1991b] rejoinds that the condition only has to hold for the $\mathbf{a}$'s such that the $\Sigma_{XX}\alpha$ project to the central subspace, and additionaly argues that the result of [Diaconis and Freedman, 1984] shows that the method will be robust to non-ellipticity for high dimensional data. [Cook and Weisberg, 1991] and [Li, 1992] note that the directions found by SIR lie in the central subspace but need not span it: for example, $Y = X_1^2 + \epsilon$ will give $E[X|Y] = 0$ despite the fact that the central subspace is spanned by x_1. [Cook and Weisberg, 1991] propose SAVE (Sliced Average Variance Estimate) to help address this, in which $var(X|Y)$ rather than $E[X|Y]$ is expanded in terms of the central subspace directions, and they also propose a more robust permutation test for statistical significance, rather than using the smallest eigienvalues of the PCA as proposed in [Li, 1991a]. [Li, 1992] also proposes a second order method, "principal Hessian directions" (pHd), to handle the symmetry problem. The idea is based on the fact that the Hessian of the forward regression function will be degenerate along directions orthogonal to the central subspace. A basis is chosen in which, for the first coordinate, the average curvature of the regression function along that axis is maximal, then this is repeated for each successive coordinate. Those coordinates are then identified as central subspace directions. [Li, 1992] recommends using both SIR and pHd for any given problem, since the former tends to be more stable, but the latter handles symmetric cases. [Li, Zha and Chiaromonte, 2005] propose approaching SDR by estimating contour directions of small variations in the response; the method solves the problem of finding the full set of central subspace directions (unlike SIR and pHd), but it still assumes elliptic X, although robustness to departures from ellipticity is claimed.

2.6.2 Kernel Dimension Reduction

SIR, where applicable, has the significant advantages that is easy to implement and can handle large datasets. However it has some limitations. First, SIR, and the above methods it inspired, assume elliptically distributed data. SIR can miss finding central subspace directions, for example if the data has symmetries under which the inverse regression is invariant, and similarly, pHd can miss such directions if the corresponding coordinates only appear as variances in f. In fact, the dimension of the space that SIR finds is bounded above, for tasks in which y takes one of c discrete values, by $c - 1$, which limits its applicability to classification tasks. Kernel dimension reduction (KDR) [Fukumizu, Bach and Jordan, 2009] addresses all of these issues, and the approach is aimed directly at the defining condition for Sufficient Dimension Reduction: $Y \perp\!\!\!\perp X | \alpha^T \mathbf{x}$. Furthermore the approach is very general. We briefly summarize the ideas here and refer the reader to [Fukumizu, Bach and Jordan, 2009] for details. Associate with the random variables X and Y (where the latter is no longer restricted to be a scalar), Reproducing Kernel Hilbert Spaces (RKHSs), H_X and H_Y. In addition assign to H_X and H_Y the Lebesque measures of the probability spaces over which X and Y are defined (so that, for example, for $f_1, f_2 \in H_X$, $\langle f_1, f_2 \rangle = \int f_1(x) f_2(x) dP(X)$). Then a "cross covariance" operator $\Sigma_{YX} : H_X \to H_Y$ can be defined so that

$$\langle g, \Sigma_{YX} f \rangle = E_{XY}[(f(X) - E_X[f(X)])(g(Y) - E_Y[g(Y)])] \qquad (2.41)$$

A *conditional covariance operator* $\Sigma_{YY|X} \equiv \Sigma_{YY} - \Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY}$ is then defined[15]. Next, introduce a matrix whose d columns (if X takes values in $\mathcal{R}^d$) are orthonormal, so that BB^T is a projection operator to the subspace spanned by those columns. Let K_X, K_Y be the kernels associated with RKHS's H_X and H_Y respectively. Define a corresponding kernel over the subspace by $K^B(\mathbf{x}_1, \mathbf{x}_2) \equiv K(B^T \mathbf{x}_1, B^T \mathbf{x}_2)$. Since the Σ operators can be defined in terms of the kernels, this leads to a cross covariance operator for the subspace: $\Sigma^B_{YY|X} \equiv \Sigma_{YY} - \Sigma^B_{YX} {\Sigma^B_{XX}}^{-1} \Sigma^B_{XY}$.

[15] This is shorthand for a form in which the last term is written in terms of bounded operators between the Hilbert spaces, and is in fact well defined when Σ_{XX}^{-1} does not exist.

[Fukumizu, Bach and Jordan, 2009] then show that, subject to some weak conditions on H_X, H_Y and the probability measures, $\Sigma^B_{YY|X} \geq \Sigma_{YY|X}$ (where the inequality refers to an ordering that can be defined for self-adjoint operators), and that $\Sigma^B_{YY|X} \geq \Sigma_{YY|X} \Leftrightarrow Y \perp\!\!\!\perp X|B^T X$. Thus the conditional covariance operator for the projected space is directly related to the conditional independence quantity we are after. Finally, the authors write a sample version of the objective function, using centered Gram matrices G^B_X, G_Y of the data (we will enounter such quantities again below), as

$$\begin{aligned} Tr[G_Y(G^B_X + m\epsilon_m I_m)^{-1}] \\ \text{subject to} \;\; B^T B = \mathbf{1} \end{aligned} \qquad (2.42)$$

where m is the sample size and ϵ a regularization parameter. B is then found using gradient descent.

So far, we have not considered the case in which there is structure in the high dimensional space. For example, suppose that your 100 dimensional data actually lies on a two dimensional torus, with noise added in the remaining 98 directions. [Nilsson, Sha and Jordan 2007] solve such structured problems for the supervised (regression) case by combining KDR with the Laplacian eigenmaps approach to manifold modeling. We will cover the latter, which is an unsupervised version of dimension reduction on manifolds, in the next chapter.

We end by noting that, while KDR is appealing in how it directly solves the SDR problem, and in its freedom from the drawbacks attending the previously mentioned methods, the above optimization problem, as stated, will be intractable for large datasets (the Gram matrices are in M_{mm}; the kernels are required to be universal [Steinwart, 2001], and will in general have high rank). However as for kernel PCA, this could be addressed by subsampling, or by using the Nyström method. Recent, parallel work by [Hsing and Ren, 2009] also propose RKHS's as providing a unified framework for dimension reduction through inverse regression. Here we have not considered the issue of consistency or convergence properties of the methods: we refer the reader to the papers for details.

2.6.3 Sufficient Dimensionality Reduction

Here we briefly describe *Sufficient Dimensionality Reduction* (SDR'), a similarly named but quite different technique [Globerson and Tishby, 2003][13]. SDR' is not a supervised method. Rather than searching for a subspace that satifies Eq. (2.34), SDR' models the density $p(X)$, parameterized by y, using two-way contingency tables. X and the model parameters Y are discrete variables (the parameters are also treated as random variables), and SDR' is a dimension reduction method in the sense that the number of parameters needed to describe $p(X)$ is reduced from $|X||Y|$ to $(d+1)(|X|+|Y|)$, where d is the dimension of a feature space to which X is mapped. The key idea of SDR' is to identify feature mappings $\phi(x)$ such that the y's can be described by a small set of such features. When p is in the exponential family, such sufficient statistics can be constructed, but this is not the case otherwise: SDR' uses an information theoretic max-min framework to quantify the information about the y's that can be gleaned from the $\phi(x)$'s: hence the term "sufficient" in the designation. Although interesting, the method is not dimension reduction in the usual sense and so we will not pursue it here.

3

Manifold Modeling

In Chapter 2 we gave an example of data with a particular geometric structure which would not be immediately revealed by examining one dimensional projections in input space[1]. How, then, can such underlying structure be found? This section outlines some methods designed to accomplish this. However we first describe the Nyström method (hereafter simply abbreviated as 'Nyström'), which provides a thread linking several of the algorithms we describe.

3.1 The Nyström method

Suppose that $K \in M_n$ and that the rank of K is $r \ll n$. Nyström gives a way of approximating the eigenvectors and eigenvalues of K using those of a small submatrix A. If A has rank r, then the decomposition is exact. This is a powerful method that can be used to speed up kernel algorithms [Williams and Seeger, 2001], to efficiently extend some algorithms (described below) to out-of-sample (test) points

[1] Although in that simple example, the astute investigator would notice that all her data vectors have the same length, and conclude from the fact that the projected density is independent of projection direction that the data must be uniformly distributed on the sphere.

[Bengio et al., 2004], and in some cases, to make an otherwise infeasible algorithm feasible [Fowlkes et al., 2004]. In this section only, we adopt the notation that matrix indices refer to sizes unless otherwise stated, so that e.g. A_{mm} means that $A \in M_m$.

3.1.1 Original Nyström

The Nyström method originated as a method for approximating the solution of Fredholm integral equations of the second kind [Press et al., 1992]. Let's consider the homogeneous d-dimensional form with density $p(\mathbf{x}),\ \mathbf{x} \in \mathcal{R}^d$. This family of equations has the form

$$\int k(\mathbf{x},\mathbf{y})u(\mathbf{y})p(\mathbf{y})d\mathbf{y} = \lambda u(\mathbf{x}) \tag{3.1}$$

The integral is approximated using the quadrature rule [Press et al., 1992]

$$\lambda u(\mathbf{x}) \approx \frac{1}{m}\sum_{i=1}^{m} k(\mathbf{x},\mathbf{x}_i)u(\mathbf{x}_i) \tag{3.2}$$

which when applied to the sample points becomes a matrix equation $K_{mm}\ \mathbf{u}_m = m\lambda\mathbf{u}_m$ (with components $K_{ij} \equiv k(\mathbf{x}_i,\mathbf{x}_j)$ and $u_i \equiv u(\mathbf{x}_i)$). This eigensystem is solved, and the value of the integral at a new point $\mathbf{x}$ is approximated by using (3.2), which gives a much better approximation that using simple interpolation [Press et al., 1992]. Thus, the original Nyström method provides a way to smoothly approximate an eigenfunction u, given its values on a sample set of points. If a different number m' of elements in the sum are used to approximate the same eigenfunction, the matrix equation becomes $K_{m'm'}\mathbf{u}_{m'} = m'\lambda\mathbf{u}_{m'}$ so the corresponding eigenvalues approximately scale with the number of points chosen. Note that we have not assumed that K is symmetric or positive semidefinite; however from now on we will assume that K is positive semidefinite.

3.1.2 Exact Nyström Eigendecomposition

Suppose that a kernel matrix $\tilde{K}_{mm}$ has rank $r < m$. Since $\tilde{K}_{mm}$ is positive semidefinite it is a Gram matrix and can be written as $\tilde{K} =$

ZZ' where $Z \in M_{mr}$ and Z is also of rank r [Horn and Johnson, 1985]. Order the row vectors in Z so that the first r are linearly independent: this just reorders rows and columns in $\tilde{K}$ to give a new kernel matrix K, but in such a way that K is still a (symmetric) Gram matrix. Then the principal submatrix $A \in S_r$ of K (which itself is the Gram matrix of the first r rows of Z) has full rank. Now letting $n \equiv m - r$, write the matrix K as

$$K_{mm} \equiv \left[\begin{array}{cc} A_{rr} & B_{rn} \\ B'_{nr} & C_{nn} \end{array} \right] \tag{3.3}$$

Since A is of full rank, the r rows $\left[\begin{array}{cc} A_{rr} & B_{rn} \end{array} \right]$ are linearly independent, and since K is of rank r, the n rows $\left[\begin{array}{cc} B'_{nr} & C_{nn} \end{array} \right]$ can be expanded in terms of them, that is, there exists H_{nr} such that

$$\left[\begin{array}{cc} B'_{nr} & C_{nn} \end{array} \right] = H_{nr} \left[\begin{array}{cc} A_{rr} & B_{rn} \end{array} \right] \tag{3.4}$$

The first r columns give $H = B'A^{-1}$, and the last n columns then give $C = B'A^{-1}B$. Thus K must be of the form[2]

$$K_{mm} = \left[\begin{array}{cc} A & B \\ B' & B'A^{-1}B \end{array} \right] = \left[\begin{array}{c} A \\ B' \end{array} \right]_{mr} A_{rr}^{-1} \left[\begin{array}{cc} A & B \end{array} \right]_{rm} \tag{3.5}$$

The fact that we've been able to write K in this 'bottleneck' form suggests that it may be possible to construct the *exact* eigendecomposition of K_{mm} (for its nonvanishing eigenvalues) using the eigendecomposition of a (possibly much smaller) matrix in M_r, and this is indeed the case [Fowlkes et al., 2004]. First use the eigendecomposition of A, $A = U\Lambda U'$, where U is the matrix of column eigenvectors of A and Λ the corresponding diagonal matrix of eigenvalues, to rewrite this in the form

$$K_{mm} = \left[\begin{array}{c} U \\ B'U\Lambda^{-1} \end{array} \right]_{mr} \Lambda_{rr} \left[\begin{array}{cc} U & \Lambda^{-1}U'B \end{array} \right]_{rm} \equiv D\Lambda D' \tag{3.6}$$

[2] It's interesting that this can be used to perform 'kernel completion', that is, reconstruction of a kernel with missing values; for example, suppose K has rank 2 and that its first two rows (and hence columns) are linearly independent, and suppose that K has met with an unfortunate accident that has resulted in all of its elements, except those in the first two rows or columns, being set equal to zero. Then the original K is easily regrown using $C = B'A^{-1}B$.

This would be exactly what we want (dropping all eigenvectors whose eigenvalues vanish), if the columns of D were orthogonal, but in general they are not. It is straightforward to show that, if instead of diagonalizing A we diagonalize $Q_{rr} \equiv A + A^{-1/2}BB'A^{-1/2} \equiv U_Q \Lambda_Q U_Q'$, then the desired matrix of orthogonal column eigenvectors is

$$V_{mr} \equiv \begin{bmatrix} A \\ B' \end{bmatrix} A^{-1/2} U_Q \Lambda_Q^{-1/2} \tag{3.7}$$

(so that $K_{mm} = V \Lambda_Q V'$ and $V'V = \mathbf{1}_{rr}$) [Fowlkes et al., 2004].

Although this decomposition is exact, this last step comes at a price: to obtain the correct eigenvectors, we had to perform an eigendecomposition of the matrix Q which depends on B. If our intent is to use this decomposition in an algorithm in which B changes when new data is encountered (for example, an algorithm which requires the eigendecomposition of a kernel matrix constructed from both train and test data), then we must recompute the decomposition each time new test data is presented. If instead we'd like to compute the eigendecomposition just once, we must approximate.

3.1.3 Approximate Nyström Eigendecomposition

Two kinds of approximation naturally arise. The first occurs if K is only approximately low rank, that is, its spectrum decays rapidly, but not to exactly zero. In this case, $B'A^{-1}B$ will only approximately equal C above, and the approximation can be quantified as $\|C - B'A^{-1}B\|$ for some matrix norm $\|\cdot\|$, where the difference is known as the Schur complement of A for the matrix K [Golub and Van Loan, 1996].

The second kind of approximation addresses the need to compute the eigendecomposition just once, to speed up test phase. The idea is simply to take Equation (3.2), sum over d elements on the right hand side where $d \ll m$ and $d > r$, and approximate the eigenvector of the full kernel matrix K_{mm} by evaluating the left hand side at all m points [Williams and Seeger, 2001]. Empirically it has been observed that choosing d to be some small integer factor larger than r works well [Platt, 2004]. How does using (3.2) correspond to the expansion in (3.6), in the case where the Schur complement vanishes? Expanding

A, B in their definition in Eq. (3.3) to A_{dd}, B_{dn}, so that U_{dd} contains the column eigenvectors of A and U_{md} contains the approximated (high dimensional) column eigenvectors, (3.2) becomes

$$U_{md}\Lambda_{dd} \approx K_{md}U_{dd} = \begin{bmatrix} A \\ B' \end{bmatrix} U_{dd} = \begin{bmatrix} U\Lambda_{dd} \\ B'U_{dd} \end{bmatrix} \tag{3.8}$$

so multiplying by Λ_{dd}^{-1} from the right shows that the approximation amounts to taking the matrix D in (3.6) as the approximate column eigenvectors: in this sense, the approximation amounts to dropping the requirement that the eigenvectors be exactly orthogonal.

We end with the following observation [Williams and Seeger, 2001]: the expression for computing the projections of a mapped test point along principal components in a kernel feature space is, apart from proportionality constants, exactly the expression for the approximate eigenfunctions evaluated at the new point, computed according to (3.2). Thus the computation of the kernel PCA features for a set of points can be viewed as using the Nyström method to approximate the full eigenfunctions at those points.

3.2 Multidimensional Scaling

We begin our look at manifold modeling algorithms with multidimensional scaling (MDS), which arose in the behavioral sciences [Borg and Groenen, 1997]. MDS starts with a measure of dissimilarity between each pair of data points in the dataset (note that this measure can be very general, and in particular can allow for non-vectorial data). Given this, MDS searches for a mapping of the (possibly further transformed) dissimilarities to a low dimensional Euclidean space such that the (transformed) pair-wise dissimilarities become squared distances. The low dimensional data can then be used for visualization, or as low dimensional features.

We start with the fundamental theorem upon which 'classical MDS' is built (in classical MDS, the dissimilarities are taken to be squared distances and no further transformation is applied [Cox and Cox, 2001]). We give a detailed proof because it will serve to illustrate a recurring theme. Let $\mathbf{e}$ be the column vector of m ones. Consider the 'centering' matrix $P^e \equiv \mathbf{1} - \frac{1}{m}\mathbf{e}\mathbf{e}'$. Let X be the matrix whose rows are the datapoints $\mathbf{x} \in \mathcal{R}^n$, $X \in M_{mn}$. Since $\mathbf{e}\mathbf{e}' \in M_m$ is the matrix of all ones, $P^e X$ subtracts the mean vector from each row $\mathbf{x}$ in X (hence the name 'centering'), and in addition, $P^e\mathbf{e} = 0$. In fact $\mathbf{e}$ is the only eigenvector (up to scaling) with eigenvalue zero, for suppose $P^e\mathbf{f} = 0$ for some $\mathbf{f} \in \mathcal{R}^m$. Then each component of $\mathbf{f}$ must be equal to the mean of all the components of $\mathbf{f}$, so all components of $\mathbf{f}$ are equal. Hence P^e has rank $m-1$, and P^e projects onto the subspace $\mathcal{R}^{m-1}$ orthogonal to $\mathbf{e}$.

By a 'distance matrix' we will mean a matrix whose ij'th element is $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ for some $\mathbf{x}_i,\ \mathbf{x}_j \in \mathcal{R}^d$, for some d, where $\|\cdot\|$ is the Euclidean norm. Notice that the elements are squared distances, despite the name. P^e can also be used to center both Gram matrices and distance matrices. We can see this as follows. Let $[C(i,j)]$ be that matrix whose ij'th element is $C(i,j)$. Then $P^e[\mathbf{x}_i \cdot \mathbf{x}_j]P^e = P^e X X' P^e = (P^e X)(P^e X)' = [(\mathbf{x}_i - \boldsymbol{\mu}) \cdot (\mathbf{x}_j - \boldsymbol{\mu})]$. In addition, using this result, $P^e[\|\mathbf{x}_i - \mathbf{x}_j\|^2]P^e = P^e[\|\mathbf{x}_i\|^2 e_i e_j + \|\mathbf{x}_j\|^2 e_i e_j - 2\mathbf{x}_i \cdot \mathbf{x}_j]P^e = -2P^e\mathbf{x}_i \cdot \mathbf{x}_j P^e = -2[(\mathbf{x}_i - \boldsymbol{\mu}) \cdot (\mathbf{x}_j - \boldsymbol{\mu})]$.

For the following theorem, the earliest form of which is due to Schoenberg [Schoenberg, 1935], we first note that, for any $A \in M_m$, and letting $Q \equiv \frac{1}{m}\mathbf{e}\mathbf{e}'$,

$$P^e A P^e = \{(\mathbf{1} - Q)A(\mathbf{1} - Q)\}_{ij} = A_{ij} - A^R_{ij} - A^C_{ij} + A^{RC}_{ij} \qquad (3.9)$$

where $A^C \equiv AQ$ is the matrix A with each column replaced by the column mean, $A^R \equiv QA$ is A with each row replaced by the row mean, and $A^{RC} \equiv QAQ$ is A with every element replaced by the mean of all the elements.

Theorem 3.1. Consider the class of symmetric matrices $A \in S_n$ such that $A_{ij} \geq 0$ and $A_{ii} = 0\ \ \forall i,j$. Then $\bar{A} \equiv -P^e A P^e$ is positive semidefinite if and only if A is a distance matrix (with embedding space $\mathcal{R}^d$ for some d). Given that A is a distance matrix, the minimal embedding dimension d is the rank of $\bar{A}$, and the embedding vectors are any set of Gram vectors of $\bar{A}$, scaled by a factor of $\frac{1}{\sqrt{2}}$.

Proof. Assume that $A \in S_m$, $A_{ij} \geq 0$ and $A_{ii} = 0 \;\; \forall i$, and that $\bar{A}$ is positive semidefinite. Since $\bar{A}$ is positive semidefinite it is also a Gram matrix, that is, there exist vectors $\mathbf{x}_i \in \mathcal{R}^m$, $i = 1, \cdots, m$ such that $\bar{A}_{ij} = \mathbf{x}_i \cdot \mathbf{x}_j$. Introduce $\mathbf{y}_i = \frac{1}{\sqrt{2}}\mathbf{x}_i$. Then from Eq. (3.9),

$$\bar{A}_{ij} = (-P^e A P^e)_{ij} = \mathbf{x}_i \cdot \mathbf{x}_j = -A_{ij} + A^R_{ij} + A^C_{ij} - A^{RC}_{ij} \qquad (3.10)$$

so that

$$\begin{aligned} 2(\mathbf{y}_i - \mathbf{y}_j)^2 \equiv (\mathbf{x}_i - \mathbf{x}_j)^2 &= A^R_{ii} + A^C_{ii} - A^{RC}_{ii} + A^R_{jj} + A^C_{jj} - A^{RC}_{jj} \\ &\quad -2(-A_{ij} + A^R_{ij} + A^C_{ij} - A^{RC}_{ij}) \\ &= 2A_{ij} \end{aligned} \qquad (3.11)$$

using $A_{ii} = 0$, $A^R_{ij} = A^R_{jj}$, $A^C_{ij} = A^C_{ii}$, and from the symmetry of A, $A^R_{ij} = A^C_{ji}$. Thus A is a distance matrix with embedding vectors $\mathbf{y}_i$. Now consider a matrix $A \in S_n$ that is a distance matrix, so that $A_{ij} = (\mathbf{y}_i - \mathbf{y}_j)^2$ for some $\mathbf{y}_i \in \mathcal{R}^d$ for some d, and let Y be the matrix whose rows are the $\mathbf{y}_i$. Then since each row and column of P^e sums to zero, we have $\bar{A} = -(P^e A P^e) = 2(P^e Y)(P^e Y)'$, hence $\bar{A}$ is positive semidefinite. Finally, given a distance matrix $A_{ij} = (\mathbf{y}_i - \mathbf{y}_j)^2$, we wish to find the dimension of the minimal embedding Euclidean space. First note that we can assume that the $\mathbf{y}_i$ have zero mean ($\sum_i \mathbf{y}_i = 0$), since otherwise we can subtract the mean from each $\mathbf{y}_i$ without changing A. Then $\bar{A}_{ij} = \mathbf{x}_i \cdot \mathbf{x}_j$, again introducing $\mathbf{x}_i \equiv \sqrt{2}\mathbf{y}_i$, so the embedding vectors $\mathbf{y}_i$ are a set of Gram vectors of $\bar{A}$, scaled by a factor of $\frac{1}{\sqrt{2}}$. Now let r be the rank of $\bar{A}$. Since $\bar{A} = XX'$, and since $\mathit{rank}(XX') = \mathit{rank}(X)$ for any real matrix X [Horn and Johnson, 1985], and since $\mathit{rank}(X)$ is the number of linearly independent $\mathbf{x}_i$, the minimal embedding space for the $\mathbf{x}_i$ (and hence for the $\mathbf{y}_i$) has dimension r. □

3.2.1 General Centering

Is P^e the most general matrix that will convert a distance matrix into a matrix of dot products? Since the embedding vectors are not unique (given a set of Gram vectors, any global orthogonal matrix applied to that set gives another set that generates the same positive semidefinite matrix), it's perhaps not surprising that the answer is no. A distance matrix is an example of a conditionally negative definite (CND) matrix. A CND matrix $D \in S_m$ is a symmetric matrix that satisfies $\sum_{i,j} a_i a_j D_{ij} \leq 0 \; \forall \{a_i \in \mathcal{R} : \sum_i a_i = 0\}$; the class of CND matrices is a superset of the class of negative semidefinite matrices [Berg et al., 1984]. Defining the projection matrix $P^c \equiv (\mathbf{1} - \mathbf{e}\mathbf{c}')$, for any $\mathbf{c} \in \mathcal{R}^m$ such that $\mathbf{e}'\mathbf{c} = 1$, then for any CND matrix D, the matrix $-P^c D P'^c$ is positive semidefinite (and hence a dot product matrix) [Schölkopf, 2001, Berg et al., 1984] (note that P^c is not necessarily symmetric). This is straightforward to prove: for any $\mathbf{z} \in \mathcal{R}^m$, $P'^c\mathbf{z} = (\mathbf{1} - \mathbf{c}\mathbf{e}')\mathbf{z} = \mathbf{z} - \mathbf{c}(\sum_a z_a)$, so $\sum_i (P'^c\mathbf{z})_i = 0$, hence $(P'^c\mathbf{z})' D (P'^c\mathbf{z}) \leq 0$ from the definition of CND. Hence we can map a distance matrix D to a dot product matrix K by using P^c in the above manner for any set of numbers c_i that sum to unity.

3.2.2 Constructing the Embedding

To actually find the embedding vectors for a given distance matrix, we need to know how to find a set of Gram vectors for a positive semidefinite matrix $\bar{A}$. Let E be the matrix of column eigenvectors $\mathbf{e}^{(\alpha)}$ (labeled by α), ordered by eigenvalue λ_α, so that the first column is the principal eigenvector, and $\bar{A}E = E\Lambda$, where Λ is the diagonal matrix of eigenvalues. Then $\bar{A}_{ij} = \sum_\alpha \lambda_\alpha e_i^{(\alpha)} e_j^{(\alpha)}$. The rows of E form the dual (orthonormal) basis to $e_i^{(\alpha)}$, which we denote $\tilde{e}_\alpha^{(i)}$. Then we can write $\bar{A}_{ij} = \sum_\alpha (\sqrt{\lambda_\alpha}\tilde{e}_\alpha^{(i)})(\sqrt{\lambda_\alpha}\tilde{e}_\alpha^{(i)})$. Hence the Gram vectors are just the dual eigenvectors with each component scaled by $\sqrt{\lambda_\alpha}$. Defining the matrix $\tilde{E} \equiv E\Lambda^{1/2}$, we see that the Gram vectors are just the rows of $\tilde{E}$.

If $\bar{A} \in S_n$ has rank $r \leq n$, then the final $n - r$ columns of $\tilde{E}$ will be zero, and we have directly found the r-dimensional embedding vectors that we are looking for. If $\bar{A} \in S_n$ is full rank, but the last $n - p$ eigenvalues are much smaller than the first p, then it's reasonable to approximate the i'th Gram vector by its first p components $\sqrt{\lambda_\alpha}\tilde{\mathbf{e}}_\alpha^{(i)}$, $\alpha = 1, \cdots, p$, and we have found a low dimensional approximation to the $\mathbf{y}$'s. This device - projecting to lower dimensions by lopping off the last few components of the dual vectors corresponding to the (possibly scaled) eigenvectors - is shared by MDS, Laplacian eigenmaps, and spectral clustering (see below). Just as for PCA, where the quality of the approximation can be characterized by the unexplained variance, we can characterize the quality of the approximation here by the squared residuals. Let $\bar{A}$ have rank r, and suppose we only keep the first $p \leq r$ components to form the approximate embedding vectors. Then denoting the approximation with a hat, the summed squared residuals are

$$\begin{aligned}\sum_{i=1}^{m} \|\hat{\mathbf{y}}_i - \mathbf{y}_i\|^2 &= \frac{1}{2}\sum_{i=1}^{m} \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|^2 \\ &= \frac{1}{2}\sum_{i=1}^{m}\sum_{a=1}^{p} \lambda_a \tilde{e}_a^{(i)2} + \frac{1}{2}\sum_{i=1}^{m}\sum_{a=1}^{r} \lambda_a \tilde{e}_a^{(i)2} - \sum_{i=1}^{m}\sum_{a=1}^{p} \lambda_a \tilde{e}_a^{(i)2}\end{aligned}$$

but $\sum_{i=1}^{m} \tilde{e}_a^{(i)2} = \sum_{i=1}^{m} e_i^{(a)2} = 1$, so

$$\sum_{i=1}^{m} \|\hat{\mathbf{y}}_i - \mathbf{y}_i\|^2 = \frac{1}{2}\left(\sum_{a=1}^{r} \lambda_a - \sum_{a=1}^{p} \lambda_a\right) = \sum_{a=p+1}^{r} \lambda_a \tag{3.12}$$

Thus the fraction of 'unexplained residuals' is $\sum_{a=p+1}^{r} \lambda_a / \sum_{a=1}^{r} \lambda_a$, in analogy to the fraction of 'unexplained variance' in PCA.

If the original symmetric matrix A is such that $\bar{A}$ is not positive semidefinite, then by the above theorem there exist no embedding points such that the dissimilarities are distances between points in some Euclidean space. In that case, we can proceed by adding a sufficiently large positive constant to the diagonal of $\bar{A}$, or by using the closest positive semidefinite matrix, in Frobenius norm[3], to $\bar{A}$, which is $\hat{A} \equiv \sum_{\alpha:\lambda_\alpha > 0} \lambda_\alpha \mathbf{e}^{(\alpha)} \mathbf{e}^{(\alpha)\prime}$. Methods such as classical MDS, that treat the dissimilarities themselves as (approximate) squared distances, are called metric scaling methods. A more general approach - 'non-metric scaling' - is to minimize a suitable cost function of the difference between the embedded squared distances, and some monotonic function of the dissimilarities [Cox and Cox, 2001]; this allows for dissimilarities which do not arise from a metric space; the monotonic function, and other weights which are solved for, are used to allow the dissimilarities to nevertheless be represented approximately by low dimensional

[3] The only proof I have seen for this assertion is due to Frank McSherry, Microsoft Research.

squared distances. An example of non-metric scaling is ordinal MDS, whose goal is to find points in the low dimensional space so that the distances there correctly reflect a given rank ordering of the original data points.

3.2.3 Landmark MDS

MDS is computationally expensive: since the distances matrix is not sparse, the computational complexity of the eigendecomposition is $O(m^3)$. This can be significantly reduced by using a method called Landmark MDS (LMDS) [Silva and Tenenbaum, 2002]. In LMDS the idea is to choose q points, called 'landmarks', where $q > r$ (where r is the rank of the distance matrix), but $q \ll m$, and to perform MDS on landmarks, mapping them to $\mathcal{R}^d$. The remaining points are then mapped to $\mathcal{R}^d$ using only their distances to the landmark points (so in LMDS, the only distances considered are those to the set of landmark points). As first pointed out in [Bengio et al., 2004] and explained in more detail in [Platt, 2005], LMDS combines MDS with the Nyström algorithm. Let $E \in S_q$ be the matrix of landmark distances and U (Λ) the matrix of eigenvectors (eigenvalues) of the corresponding kernel matrix $A \equiv -\frac{1}{2}P^c E P'^c$, so that the embedding vectors of the landmark points are the first d elements of the rows of $U\Lambda^{1/2}$. Now, extending E by an extra column and row to accommodate the squared distances from the landmark points to a test point, we write the extended distance matrix and corresponding kernel as

$$D = \begin{bmatrix} E & \mathbf{f} \\ \mathbf{f}' & g \end{bmatrix}, \qquad K \equiv -\frac{1}{2}P^c D P'^c = \begin{bmatrix} A & \mathbf{b} \\ \mathbf{b}' & c \end{bmatrix} \tag{3.13}$$

Then from Eq. (3.6) we see that the Nyström method gives the approximate column eigenvectors for the extended system as

$$\begin{bmatrix} U \\ \mathbf{b}' U \Lambda^{-1} \end{bmatrix} \tag{3.14}$$

Thus the embedding coordinates of the test point are given by the first d elements of the row vector $\mathbf{b}' U \Lambda^{-1/2}$. However, we only want to compute U and Λ once - they must not depend on the test point. [Platt, 2005] has pointed out that this can be accomplished by choosing the centering coefficients c_i in $P^c \equiv \mathbf{1} - \mathbf{e}\mathbf{c}'$ such that $c_i = 1/q$ for $i \leq q$ and $c_{q+1} = 0$: in that case, since

$$K_{ij} = -\frac{1}{2}\left(D_{ij} - e_i(\sum_{k=1}^{q+1} c_k D_{kj}) - e_j(\sum_{k=1}^{q+1} D_{ik} c_k) + e_i e_j(\sum_{k,m=1}^{q+1} c_k D_{km} c_m)\right) \tag{3.15}$$

the matrix A (found by limiting i, j to $1, \ldots, q$ above) depends only on the matrix E above. Finally, we need to relate $\mathbf{b}$ back to the measured quantities - the vector of squared distances from the test point to the landmark points. Using $b_i = (-\frac{1}{2}P^c D P'^c)_{q+1,i}$, $i = 1, \cdots, q$, we find that

$$b_k = -\frac{1}{2}\left[D_{q+1,k} - \frac{1}{q}\sum_{j=1}^{q} D_{q+1,j} e_k - \frac{1}{q}\sum_{i=1}^{q} D_{ik} + \frac{1}{q^2}\left(\sum_{i,j=1}^{q} D_{ij}\right) e_k\right] \tag{3.16}$$

The first term in the square brackets is the vector of squared distances from the test point to the landmarks, $\mathbf{f}$. The third term is the row mean of the landmark distance squared matrix, $\bar{E}$. The second and fourth terms are proportional to the vector of all ones $\mathbf{e}$, and can be dropped[4] since $U'\mathbf{e} = 0$. Hence, modulo terms which vanish when constructing the embedding coordinates, we have $\mathbf{b} \simeq -\frac{1}{2}(\mathbf{f} - \bar{E})$, and the coordinates of the embedded test point are $\frac{1}{2}\Lambda^{-1/2}U'(\bar{E} - \mathbf{f})$; this reproduces the form given in [Silva and Tenenbaum, 2002]. Landmark MDS has two significant advantages: first, it reduces the computational complexity from $O(m^3)$ to $O(q^3 + q^2(m-q) = q^2 m)$; and second, it can be applied to any non-landmark point, and so gives a method of extending MDS (using Nyström) to out-of-sample data.

3.3 Isomap

MDS is valuable for extracting low dimensional representations for some kinds of data, but it does not attempt to explicitly model the underlying manifold. Two methods that do directly model the manifold are Isomap and Locally Linear Embedding. Suppose that as in Section 2.1.1, again unbeknownst to you, your data lies on a curve,

[4] The last term can also be viewed as an unimportant shift in origin; in the case of a single test point, so can the second term, but we cannot rely on this argument for multiple test points, since the summand in the second term depends on the test point.

but in contrast to Section 2.1.1, the curve is not a straight line; in fact it is sufficiently complex that the minimal embedding space $\mathcal{R}^d$ that can contain it has high dimension d. PCA will fail to discover the one dimensional structure of your data; MDS will also, since it attempts to faithfully preserve all distances. Isomap (isometric feature map) [Tenenbaum, 1998], on the other hand, will succeed. The key assumption made by Isomap is that the quantity of interest, when comparing two points, is the distance along the curve between the two points; if that distance is large, it is to be taken, even if in fact the two points are close in $\mathcal{R}^d$ (this example also shows that noise must be handled carefully). The low dimensional space can have more than one dimension: [Tenenbaum, 1998] gives an example of a 5 dimensional manifold embedded in a 50 dimensional space. The basic idea is to construct a graph whose nodes are the data points, where a pair of nodes are adjacent only if the two points are close in $\mathcal{R}^d$, and then to approximate the geodesic distance along the manifold between any two points as the shortest path in the graph, computed using the Floyd algorithm [Gondran and Minoux, 1984]; and finally to use MDS to extract the low dimensional representation (as vectors in $\mathcal{R}^{d'}$, $d' \ll d$) from the resulting matrix of squared distances ([Tenenbaum, 1998] suggests using ordinal MDS, rather than metric MDS, for robustness).

Isomap shares with the other manifold mapping techniques we describe the property that it does not provide a direct functional form for the mapping $\mathcal{I} : \mathcal{R}^d \rightarrow \mathcal{R}^{d'}$ that can simply be applied to new data, so computational complexity of the algorithm is an issue in test phase. The eigenvector computation is $O(m^3)$, and the Floyd algorithm also $O(m^3)$, although the latter can be reduced to $O(hm^2 \log m)$ where h is a heap size [Silva and Tenenbaum, 2002]. Landmark Isomap simply employs landmark MDS [Silva and Tenenbaum, 2002] to addresses this problem, computing all distances as geodesic distances to the landmarks. This reduces the computational complexity to $O(q^2m)$ for the LMDS step, and to $O(hqm \log m)$ for the shortest path step.

3.4 Locally Linear Embedding

Locally linear embedding (LLE) [Roweis and Saul, 2000] models the manifold by treating it as a union of linear patches, in analogy to using coordinate charts to parameterize a manifold in differential geometry. Suppose that each point $\mathbf{x}_i \in \mathcal{R}^d$ has a small number of close neighbours indexed by the set $\mathcal{N}(i)$, and let $\mathbf{y}_i \in \mathcal{R}^{d'}$ be the low dimensional representation of $\mathbf{x}_i$. The idea is to express each $\mathbf{x}_i$ as a linear combination of its neighbours, and then construct the $\mathbf{y}_i$ so that they can be expressed as the same linear combination of their corresponding neighbours (the latter also indexed by $\mathcal{N}(i)$). To simplify the discussion let's assume that the number of the neighbours is fixed to n for all i. The condition on the $\mathbf{x}$'s can be expressed as finding that $W \in M_{mn}$ that minimizes the sum of the reconstruction errors, $\sum_i \|\mathbf{x}_i - \sum_{j \in \mathcal{N}(i)} W_{ij}\mathbf{x}_j\|^2$. Each reconstruction error $E_i \equiv \|\mathbf{x}_i - \sum_{j \in \mathcal{N}(i)} W_{ij}\mathbf{x}_j\|^2$ should be unaffected by any global translation $\mathbf{x}_i \to \mathbf{x}_i + \boldsymbol{\delta}$, $\boldsymbol{\delta} \in \mathcal{R}^d$, which gives the condition $\sum_{j \in \mathcal{N}(i)} W_{ij} = 1 \;\forall i$. Note that each E_i is also invariant to global rotations and reflections of the coordinates. Thus the objective function we wish to minimize is

$$F \equiv \sum_i F_i \equiv \sum_i \left(\frac{1}{2}\|\mathbf{x}_i - \sum_{j \in \mathcal{N}(i)} W_{ij}\mathbf{x}_j\|^2 - \lambda_i \left(\sum_{j \in \mathcal{N}(i)} W_{ij} - 1 \right) \right)$$

where the constraints are enforced with Lagrange multipliers λ_i [Burges, 2004]. Since the sum splits into independent terms we can minimize each F_i separately. Thus fixing i and letting $\mathbf{x} \equiv \mathbf{x}_i$, $\mathbf{v} \in \mathcal{R}^n$, $v_j \equiv W_{ij}$, and $\lambda \equiv \lambda_i$, and introducing the matrix $C \in S_n$, $C_{jk} \equiv \mathbf{x}_j \cdot \mathbf{x}_k$, $j, k \in \mathcal{N}(i)$, and the vector $\mathbf{b} \in \mathcal{R}^n$, $b_j \equiv \mathbf{x} \cdot \mathbf{x}_j$, $j \in \mathcal{N}(i)$, then requiring that the derivative of F_i with respect to v_j vanishes gives $\mathbf{v} = C^{-1}(\lambda\mathbf{e} + \mathbf{b})$. Imposing the constraint $\mathbf{e}'\mathbf{v} = 1$ then gives $\lambda = (1 - \mathbf{e}'C^{-1}\mathbf{b})/(\mathbf{e}'C^{-1}\mathbf{e})$. Thus W can be found by applying this for each i.

Given the W's, the second step is to find a set of $\mathbf{y}_i \in \mathcal{R}^{d'}$ that can be expressed in terms of each other in the same manner. Again no exact solution may exist and so $\sum_i \|\mathbf{y}_i - \sum_{j \in \mathcal{N}(i)} W_{ij}\mathbf{y}_j\|^2$ is minimized with respect to the $\mathbf{y}$'s, keeping the W's fixed. Let $Y \in M_{md'}$ be the matrix of row vectors of the points $\mathbf{y}$. [Roweis and Saul, 2000] enforce the condition that the $\mathbf{y}$'s span a space of dimension d' by requiring that $(1/m)Y'Y = \mathbf{1}$, although any condition of the form $Y'PY = Z$ where $P \in S_m$ and $Z \in S_{d'}$ is of full rank would suffice (see Section 3.5.1). The origin is arbitrary; the corresponding degree of freedom can be removed by requiring that the $\mathbf{y}$'s have zero mean, although in fact this need not be explicitly imposed as a constraint on the optimization, since the set of solutions can easily be chosen to have this property. The rank constraint requires that the $\mathbf{y}$'s have unit covariance; this links the variables so that the optimization no longer decomposes into m separate optimizations: introducing Lagrange multipliers $\lambda_{\alpha\beta}$ to enforce the constraints, the objective function to be minimized is

$$F = \frac{1}{2}\sum_i \|\mathbf{y}_i - \sum_j W_{ij}\mathbf{y}_j\|^2 - \frac{1}{2}\sum_{\alpha\beta}\lambda_{\alpha\beta}\left(\sum_i \frac{1}{m}Y_{i\alpha}Y_{i\beta} - \delta_{\alpha\beta}\right) \tag{3.17}$$

where for convenience we treat the W's as matrices in M_m, where $W_{ij} \equiv 0$ for $j \notin \mathcal{N}(i)$. Taking the derivative with respect to $Y_{k\delta}$ and choosing $\lambda_{\alpha\beta} = \lambda_\alpha \delta_{\alpha\beta} \equiv \Lambda_{\alpha\beta}$ gives[18] the matrix equation

$$(\mathbf{1} - W)'(\mathbf{1} - W)Y = \frac{1}{m}Y\Lambda \tag{3.18}$$

Since $(\mathbf{1} - W)'(\mathbf{1} - W) \in S_m$, its eigenvectors are, or can be chosen to be, orthogonal; and since $(\mathbf{1} - W)'(\mathbf{1} - W)\mathbf{e} = 0$, choosing the columns of Y to be the next d' eigenvectors of $(1 - W)'(1 - W)$ with the smallest eigenvalues guarantees that the $\mathbf{y}$ are zero mean (since they are orthogonal to $\mathbf{e}$). We can also scale the $\mathbf{y}$ so that the columns of Y are orthonormal, thus satisfying the covariance constraint $Y'Y = \mathbf{1}$. Finally, the lowest-but-one weight eigenvectors are chosen because their corresponding eigenvalues sum to $m\sum_i \|\mathbf{y}_i - \sum_j W_{ij}\mathbf{y}_j\|^2$, as can be seen by applying Y' to the left of (3.18).

Thus, LLE requires a two-step procedure. The first step (finding the W's) has $O(n^3 m)$ computational complexity; the second requires eigendecomposing the product of two sparse matrices in M_m. LLE has the desirable property that it will result in the same weights W if the data is scaled, rotated, translated and / or reflected.

3.5 Graphical Methods

In this section we review two interesting methods that connect with spectral graph theory. Let's start by defining a simple mapping from a dataset to an undirected graph G by forming a one-to-one correspondence between nodes in the graph and data points. If two nodes i, j are connected by an arc, associate with it a positive arc weight W_{ij}, $W \in S_m$, where W_{ij} is a similarity measure between points $\mathbf{x}_i$ and $\mathbf{x}_j$. The arcs can be defined, for example, by the minimum spanning tree, or by forming the N nearest neighbours, for N sufficiently large. The Laplacian matrix for any weighted, undirected graph is defined [Chung, 1997] by $\mathcal{L} \equiv D^{-1/2}LD^{-1/2}$, where $L_{ij} \equiv D_{ij} - W_{ij}$ and where $D_{ij} \equiv \delta_{ij}(\sum_k W_{ik})$. We can see that $\mathcal{L}$ is positive semidefinite as follows: for any vector $\mathbf{z} \in \mathcal{R}^m$, since $W_{ij} \geq 0$,

$$0 \leq \frac{1}{2}\sum_{i,j}(z_i - z_j)^2 W_{ij} = \sum_i z_i^2 D_{ii} - \sum_{i,j} z_i W_{ij} z_j = \mathbf{z}' L \mathbf{z}$$

and since L is positive semidefinite, so is the Laplacian. Note that L is never positive definite since the vector of all ones, $\mathbf{e}$, is always an eigenvector with eigenvalue zero (and similarly $\mathcal{L}D^{1/2}\mathbf{e} = 0$).

Let G be a graph and m its number of nodes. For $W_{ij} \in \{0, 1\}$, the spectrum of G (defined as the set of eigenvalues of its Laplacian) characterizes its global properties [Chung, 1997]: for example, a complete graph (that is, one for which every node is adjacent to every other node) has a single zero eigenvalue, and all other eigenvalues are equal to $\frac{m}{m-1}$; if G is connected but not complete, its smallest nonzero eigenvalue is bounded above by unity; the number of zero eigenvalues is equal to the number of connected components in the graph, and in fact the spectrum of a graph is the union of the spectra of its connected components; and the sum of the eigenvalues is bounded above by m, with equality iff G has no isolated nodes. In light of these results, it seems reasonable to expect that global properties of the data - how it clusters, or what dimension manifold it lies on - might be captured by properties of the Laplacian. The following two approaches leverage this idea. We note that using similarities in this manner results in local algorithms: since each node is only adjacent to a small set of similar nodes, the resulting matrices are sparse and can therefore be eigendecomposed efficiently.

3.5.1 Laplacian Eigenmaps

The Laplacian eigenmaps algorithm [Belkin and Niyogi, 2003] uses $W_{ij} = \exp^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2}$. Let $\mathbf{y}(\mathbf{x}) \in \mathcal{R}^{d'}$ be the embedding of sample vector $\mathbf{x} \in \mathcal{R}^d$, and let $Y_{ij} \in M_{md'} \equiv (\mathbf{y}_i)_j$. We would like to find $\mathbf{y}$'s that minimize $\sum_{i,j} \|\mathbf{y}_i - \mathbf{y}_j\|^2 W_{ij}$, since then if two points are similar, their $\mathbf{y}$'s will be close, whereas if $W \approx 0$, no restriction is put on their $\mathbf{y}$'s. We have:

$$\sum_{i,j} \|\mathbf{y}_i - \mathbf{y}_j\|^2 W_{ij} = 2 \sum_{i,j,a} (\mathbf{y}_i)_a (\mathbf{y}_j)_a (D_{ii}\delta_{ij} - W_{ij}) = 2\mathrm{Tr}(Y'LY) \tag{3.19}$$

In order to ensure that the target space has dimension d' (minimizing (3.19) alone has solution $Y = 0$), we require that Y have rank d. Any constraint of the form $Y'PY = Z$, where $P \in S_m$ and $m \geq d'$, will suffice, provided that $Z \in S_{d'}$ is of full rank. This can be seen as follows: since the rank of Z is d' and since the rank of a product of matrices is bounded above by the rank of each, we have that $d' = rank(Z) = rank(Y'PY) \leq \min(rank((Y'), rank(P),\ rank(Y))$, and so $rank(Y) \geq d'$; but since $Y \in M_{md'}$ and $d' \leq m$, the rank of Y is at most d'; hence $rank(Y) = d'$. However, minimizing $\mathrm{Tr}(Y'LY)$ subject to the constraint $Y'DY = \mathbf{1}$ results in the simple generalized eigenvalue problem $L\mathbf{y} = \lambda D\mathbf{y}$ [Belkin and Niyogi, 2003]. It's useful to see how this arises: we wish to minimize $\mathrm{Tr}(Y'LY)$ subject to the $d'(d'+1)/2$ constraints $Y'DY = \mathbf{1}$. Let $a, b = 1, \ldots, d$ and $i, j = 1, \ldots, m$. Introducing (symmetric) Lagrange multipliers λ_{ab} leads to the objective function $\sum_{i,j,a} y_{ia} L_{ij} y_{ja} - \sum_{i,j,a,b} \lambda_{ab}(y_{ia} D_{ij} y_{jb} - \delta_{ab})$, with extrema at $\sum_j L_{kj} y_{j\beta} = \sum_{\alpha,i} \lambda_{\alpha\beta} D_{ki} y_{i\alpha}$. We choose[5] $\lambda_{\alpha\beta} \equiv \lambda_\beta \delta_{\alpha\beta}$, giving $\sum_j L_{kj} y_{j\alpha} = \sum_i \lambda_\alpha D_{ki} y_{i\alpha}$. This is a generalized eigenvector problem with eigenvectors the columns of Y. Hence once again the low dimensional vectors are constructed from the first few components of the dual eigenvectors, except that in this case, the eigenvectors with lowest eigenvalues are chosen (omitting the eigenvector $\mathbf{e}$), and in contrast to

[5] This same trick was used in the previous Chapter: Lagrange multipliers can be chosen in any way that results in a solution satisfying the constraints.

MDS, they are not weighted by the square roots of the eigenvalues. Thus Laplacian eigenmaps must use some other criterion for deciding on what d' should be. Finally, note that the $\mathbf{y}$'s are conjugate with respect to D (as well as L), so we can scale them so that the constraints $Y'DY = \mathbf{1}$ are indeed met, and our drastic simplification of the Lagrange multipliers did no damage; and left-multiplying the eigenvalue equation by $\mathbf{y}'_\alpha$ shows that $\lambda_\alpha = \mathbf{y}'_\alpha L \mathbf{y}_\alpha$, so choosing the smallest eigenvalues indeed gives the lowest values of the objective function, subject to the constraints.

3.5.2 Spectral Clustering

Although spectral clustering is a clustering method, it is very closely related to dimension reduction. In fact, since clusters may be viewed as large scale structural features of the data, any dimension reduction technique that maintains these structural features will be a good preprocessing step prior to clustering, to the point where very simple clustering algorithms (such as K-means) on the preprocessed data can work well [Shi and Malik, 2000, Meila and Shi, 2000, Ng et al., 2002]. If a graph is partitioned into two disjoint sets by removing a set of arcs, the *cut* is defined as the sum of the weights of the removed arcs. Given the mapping of data to graph defined above, a cut defines a split of the data into two clusters, and the minimum cut encapsulates the notion of maximum dissimilarity between two clusters. However finding a minimum cut tends to just lop off outliers, so [Shi and Malik, 2000] define a normalized cut, which is now a function of all the weights in the graph, but which penalizes cuts which result in a subgraph g such that the cut divided by the sum of weights from g to G is large; this solves the outlier problem. Now suppose we wish to divide the data into two clusters. Define a scalar on each node, $z_i,\ i = 1, \ldots, m$, such that $z_i = 1$ for nodes in one cluster and $z_i = -1$ for nodes in the other. The solution to the normalized mincut problem is given by [Shi and Malik, 2000]

$$\min_{\mathbf{y}} \frac{\mathbf{y}'L\mathbf{y}}{\mathbf{y}'D\mathbf{y}} \text{ such that } y_i \in \{1, -b\} \text{ and } \mathbf{y}'D\mathbf{e} = 0 \qquad (3.20)$$

where $\mathbf{y} \equiv (\mathbf{e} + \mathbf{z}) + b(\mathbf{e} - \mathbf{z})$, and b is a constant that depends on the partition. This problem is solved by relaxing $\mathbf{y}$ to take real values: the problem then becomes finding the second smallest eigenvector of the generalized eigenvalue problem $L\mathbf{y} = \lambda D\mathbf{y}$ (the constraint $\mathbf{y}'D\mathbf{e} = 0$ is automatically satisfied by the solutions), which is exactly the same problem found by Laplacian eigenmaps (in fact the objective function used by Laplacian eigenmaps was proposed as Eq. (10) in [Shi and Malik, 2000]). The algorithms differ in what they do next. The clustering is achieved by thresholding the element y_i so that the nodes are split into two disjoint sets. The dimension reduction is achieved by treating the element y_i as the first component of a reduced dimension representation of the sample $\mathbf{x}_i$. There is also an interesting equivalent physical interpretation, where the arcs are springs, the nodes are masses, and the $\mathbf{y}$ are the fundamental modes of the resulting vibrating system [Shi and Malik, 2000]. [Meila and Shi, 2000] point out that that matrix $P \equiv D^{-1}L$ is stochastic, which motivates the interpretation of spectral clustering as the stationary distribution of a Markov random field: the intuition is that a random walk, once in one of the mincut clusters, tends to stay in it. The stochastic interpretation also provides tools to analyse the thresholding used in spectral clustering, and a method for learning the weights W_{ij} based on training data with known clusters [Meila and Shi, 2000]. The dimension reduction view also motivates a different approach to clustering, where instead of simply clustering by thresholding a single eigenvector, simple clustering algorithms are applied to the low dimensional representation of the data [Ng et al., 2002].

3.6 Pulling the Threads Together

At this point the reader is probably struck by how similar the mathematics underlying all of these approaches is. We've used essentially the same Lagrange multiplier trick to enforce constraints three times; all of the methods in this Chapter (and most in this review) rely heavily on eigendecompositions. Isomap, LLE, Laplacian eigenmaps, and spectral clustering all share the property that in their original forms, they do not provide a direct functional form for the dimension-reducing mapping, so the extension to new data requires re-training. Landmark Isomap solves this problem; the other algorithms could also use Nyström to solve it (as pointed out by [Bengio et al., 2004]). Isomap is often called a 'global' dimension reduction algorithm, because it attempts to preserve all geodesic distances; by contrast, LLE, spectral clustering and Laplacian eigenmaps are local (for example, LLE attempts to preserve local translations, rotations and scalings of the data). Landmark Isomap is still global in this sense, but the landmark device brings the computational cost more in line with the other algorithms. Although they start from different geometrical considerations, LLE, Laplacian eigenmaps, spectral clustering and MDS all look quite similar under the hood: the first three use the dual eigenvectors of a symmetric matrix as their low dimensional representation, and MDS uses the dual eigenvectors with components scaled by square roots of eigenvalues. In light of this it's perhaps not surprising that relations linking these algorithms can be found: for example, given certain assumptions on the smoothness of the eigenfunctions and on the distribution of the data, the eigendecomposition performed by LLE can be shown to coincide with the eigendecomposition of the squared Laplacian [Belkin and Niyogi, 2003]; and [Ham et al., 2004] show how Laplacian eigenmaps, LLE and Isomap can be viewed as variants of kernel PCA. [Platt, 2005] links several flavors of MDS by showing how landmark MDS and two other MDS algorithms (not described here) are in fact all Nyström algorithms. Despite the mathematical similarities of LLE, Isomap and Laplacian Eigenmaps, their different geometrical roots result in different properties: for example, for data which lies on a manifold of dimension d embedded in a higher dimensional space, the eigenvalue spectrum of the LLE and Laplacian Eigenmaps algorithms do not reveal anything about d, whereas the spectrum for Isomap (and MDS) does.

The connection between MDS and PCA goes further than the form taken by the 'unexplained residuals' in Eq. (3.12). If $X \in M_{md}$ is the matrix of m (zero-mean) sample vectors, then PCA diagonalizes the covariance matrix $X'X$, whereas MDS diagonalizes the kernel matrix XX'; but XX' has the same eigenvalues as $X'X$ [Horn and Johnson, 1985], and $m - d$ additional zero eigenvalues (if $m > d$). In fact if $\mathbf{v}$ is an eigenvector of the kernel matrix so that $XX'\mathbf{v} = \lambda\mathbf{v}$, then clearly $X'X(X'\mathbf{v}) = \lambda(X'\mathbf{v})$, so $X'\mathbf{v}$ is an eigenvector of the covariance matrix, and similarly if $\mathbf{u}$ is an eigenvector of the covariance matrix, then $X\mathbf{u}$ is an eigenvector of the kernel matrix. This provides one way to view how kernel PCA computes the eigenvectors of the (possibly infinite dimensional) covariance matrix in feature space in terms of the eigenvectors of the kernel matrix. There's a useful lesson here: given a covariance matrix (Gram matrix) for which you wish to compute those eigenvectors with nonvanishing eigenvalues, and if the corresponding Gram matrix (covariance matrix) is both available, and more easily eigendecomposed (has fewer elements), then compute the eigenvectors for the latter, and map to the eigenvectors of the former using the data matrix as above. Along these lines, [Williams, 2001] has pointed out that kernel PCA can itself be viewed as performing MDS in feature space. Before kernel PCA is performed, the kernel is centered (i.e. $P^e K P^e$ is computed), and for kernels that depend on the data only through functions of squared distances between points (such as radial basis function kernels), this centering is equivalent to centering a distance matrix in feature space. [Williams, 2001] further points out that for these kernels, classical MDS in feature space is equivalent to a form of metric MDS in input space. Although ostensibly kernel PCA gives a function that can be applied to test points, while MDS does not, kernel PCA does so by using the Nyström approximation (see Section 3.1.3), and exactly the same can be done with MDS.

4

Conclusion

Dimension reduction has benefited from a great deal of work in both the statistics and machine learning communities. In this review I've limited the scope largely to geometric methods, so I'd like to alert the reader to three other interesting leads. The first is the method of principal curves, where the idea is to find that smooth curve that passes through the data in such a way that the sum of shortest distances from each point to the curve is minimized, thus providing a nonlinear, one-dimensional summary of the data [Hastie and Stuetzle, 1989]; the idea has since been extended by applying various regularization schemes (including kernel-based), and to manifolds of higher dimension [Schölkopf and Smola, 2002]. Second, the Information Bottleneck method [Tishby et al., 1999], which aims to distill the information in a random (predictor) variable X that is needed to describe a (response) variable Y, using a model variable Z, maximizes the (weighted) difference in mutual information between Y and Z, and between X and Z. The use of information theory is intuitively appealing, although the method requires that the joint density $p(X, Y)$ be estimated. Finally, we note that the subject of feature selection, where one wants to select an optimal subset of features (for example, in order to build smaller, faster models), is a restricted form of dimension reduction. Along those lines, [Viola and Jones, 2001] showed that boosted stump models can be very effective for finding a small set of good features from a large set of possible features. For more on feature selection, we refer the reader to [Guyon, 2003].

Acknowledgements

Thanks to Michael Jordan, John Platt, and Dennis Cook, for valuable comments and suggestions.

References

[Aizerman et al., 1964] M.A. Aizerman, E.M. Braverman, and L.I. Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

[Bach and Jordan, 2002] F. R. Bach and M. I. Jordan. Kernel Independent Component Analysis. *Journal of machine learning research*, 3:1–48, 2002

[Baldi and Hornik, 1995] P.F. Baldi and K. Hornik. Learning in linear neural networks: A survey. *IEEE Transactions on Neural Networks*, 6(4):837–858, July 1995.

[Basilevsky, 1994] A. Basilevsky. *Statistical Factor Analysis and Related Methods.* Wiley, New York, 1994.

[Belkin and Niyogi, 2003] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[Bengio et al., 2004] Y. Bengio, J. Paiement, and P. Vincent. Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps and spectral clustering. In *Advances in Neural Information Processing Systems 16*. MIT Press, 2004.

[Berg et al., 1984] C. Berg, J.P.R. Christensen, and P. Ressel. *Harmonic Analysys on Semigroups.* Springer-Verlag, 1984.

[Bishop, 1999] C. M. Bishop. Bayesian PCA. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11, pages 382–388, Cambridge, MA, 1999. The MIT Press.

[Borg and Groenen, 1997] I. Borg and P. Groenen. *Modern Multidimensional Scaling: Theory and Applications.* Springer, 1997.

[Boser et al., 1992] B. E. Boser, I. M. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Fifth Annual Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, 1992. ACM.

[Burges, 2004] C.J.C. Burges. Some Notes on Applied Mathematics for Machine Learning. In O. Bousquet, U. von Luxburg, and G. Rätsch, editors, *Advanced Lectures on Machine Learning*, pages 21–40. Springer Lecture Notes in Aritificial Intelligence, 2004.

[Burges, 2005] C.J.C. Burges. Geometric Methods for Feature Selection and Dimensional Reduction. *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*, Kluwer Academic, Eds. L. Rokach and O. Maimon, 2005.

[Burges et al., 2002] C.J.C. Burges, J.C. Platt, and S. Jana. Extracting noise-robust features from audio. In *Proc. IEEE Conference on Acoustics, Speech and Signal Processing*, pages 1021–1024. IEEE Signal Processing Society, 2002.

[Burges et al., 2003] C.J.C. Burges, J.C. Platt, and S. Jana. Distortion discriminant analysis for audio fingerprinting. *IEEE Transactions on Speech and Audio Processing*, 11(3):165–174, 2003.

[Chung, 1997] F.R.K. Chung. *Spectral Graph Theory.* American Mathematical Society, 1997.

[Cook and Weisberg, 1991] R.D. Cook and S. Weisberg. Sliced Inverse Regression for Dimension Reduction: Comment. *Journal of the American Statistical Association*, 86(414):328–332, 1991.

[Cook, 1998] R.D. Cook. *Regression Graphics.* Wiley, 1998.

[Cook and Lee, 1999] R.D. Cook and H. Lee. Dimension Reduction in Binary Response Regression. *Journal of the American Statistical Association*, 94 (448): 1187-1200, 1999.

[Cox and Cox, 2001] T.F. Cox and M.A.A. Cox. *Multidimensional Scaling.* Chapman and Hall, 2001.

[Darlington, 1997] R.B. Darlington. Factor analysis. Technical report, Cornell University, 1997 http://comp9.psych.cornell.edu/Darlington/factor.htm.

[Silva and Tenenbaum, 2002] V. de Silva and J.B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 705–712. MIT Press, 2002.

[Diaconis and Freedman, 1984] P. Diaconis and D. Freedman. Asymptotics of graphical projection pursuit. *Annals of Statistics*, 12:793–815, 1984.

[Diamantaras and Kung, 1996] K.I. Diamantaras and S.Y. Kung. *Principal Component Neural Networks.* John Wiley, 1996.

[Duda and Hart, 1973] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* John Wiley, 1973.

[Fowlkes et al., 2004] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nyström method. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(2), 2004.

[Friedman and Stuetzle, 1981] J.H. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of the American Statistical Association*, 76(376):817–823, 1981.

[Friedman et al., 1984] J.H. Friedman, W. Stuetzle, and A. Schroeder. Projection pursuit density estimation. *J. Amer. Statistical Assoc.*, 79:599–608, 1984.

[Friedman and Tukey, 1974] J.H. Friedman and J.W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, c-23(9):881–890, 1974.

[Fukumizu, Bach and Jordan, 2009] K. Fukumizu, F.R. Bach and M.I. Jordan. Kernel Dimension Reduction in Regression. *Annals of Statistics*, to appear, http://www.ism.ac.jp/ fukumizu/papers/KDR_AS.pdf.

[Globerson and Tishby, 2003] A. Globerson and N. Tishby. Sufficient Dimensionality Reduction. *Journal of Machine Learning Research*, 3, 2003.

[Golub and Van Loan, 1996] G.H. Golub and C.F. Van Loan. *Matrix Computations.* Johns Hopkins, third edition, 1996.

[Gondran and Minoux, 1984] M. Gondran and M. Minoux. *Graphs and Algorithms.* John Wiley and Sons, 1984.

[Grimmet and Stirzaker, 2001] G. Grimmet and D. Stirzaker. *Probability and Random Processes.* Oxford University Press, third edition, 2001.

[Guyon, 2003] I. Guyon. *NIPS 2003 workshop on feature extraction:* http://clopinet.com/isabelle/Projects/NIPS2003/.

[Ham et al., 2004] J. Ham, D.D. Lee, S. Mika, and B. Schölkopf. A kernel view of dimensionality reduction of manifolds. In *Proceedings of the International Conference on Machine Learning*, 2004.

[Hardoon, Szedmak and Shawe-Taylor, 2004] D.R. Hardoon, S. Szedmak and J. Shawe-Taylor. Canonical correlation analysis: an overview with application to learning methods. *Neural Computation*, 12(16):2639–2664, 2004.

[Hastie and Stuetzle, 1989] T.J. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84(406):502–516, 1989.

[Hsing and Ren, 2009] T. Hsing and H. Ren. An RKHS formulation of the inverse regression dimension-reduction problem. *Annals of Statistics*, 37(2):726–755, 2009

[Horn and Johnson, 1985] R.A. Horn and C.R. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

[Hotelling, 1936] H. Hotelling. Relations between two sets of variates. Biometrika, 28:321–377, 1936.

[Huber, 1985] P.J. Huber. Projection pursuit. *Annals of Statistics*, 13(2):435–475, 1985.

[Hyvärinen et al., 2001] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis.* Wiley, 2001.

[Kelly, 1928] T. L. Kelly. *Crossroads in the Mind of Man; A study of Differentiable Mental Abilities".* Stanford University Press, 1928.

[Kimeldorf and Wahba, 1971] G.S. Kimeldorf and G. Wahba. Some results on Tchebycheffian Spline Functions. *J. Mathematical Analysis and Applications*, 33:82–95, 1971.

[LeCun and Bengio, 1995] Y. LeCun and Y. Bengio. Convolutional networks for images, speech and time-series. In M. Arbib, editor, *The Handbook of Brain Theory and Neural Networks.* MIT Press, 1995.

[Li, 1991a] C-K. Li. Sliced Inverse Regression for Dimension Reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991.

[Li, 1991b] C-K. Li. Sliced Inverse Regression for Dimension Reduction: Rejoinder. *Journal of the American Statistical Association*, 86(414):337–342, 1991.

[Li, 1992] C-K. Li. On Principal Hessian Directions for Data Visualization and Dimension Reduction: Another Application of Stein's Lemma. *Journal of the American Statistical Association*, 87(420):1025–1039, 1992.

[Li, Zha and Chiaromonte, 2005] B. Li, H. Zha and F. Chiaromonte. Contour Regression: A General Approach to Dimension Reduction. *The Annals of Statistics*, 33(4):1580–1616, 2005.

[Meila and Shi, 2000] M. Meila and J. Shi. Learning segmentation by random walks. In *Advances in Neural Information Processing Systems*, pages 873–879, 2000.

[Mika et al., 1999] S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz, and G. Rätsch. Kernel PCA and de–noising in feature spaces. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems 11*. MIT Press, 1999.

[Ng et al., 2002] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.

[Nilsson, Sha and Jordan 2007] J. Nilsson and F. Sha and M.I. Jordan. Regression on Manifolds using Kernel Dimension Reduction. *Proceedings of the 24th International Conference on Machine Learning*, 2007

[Platt, 2004] J. Platt. *Private Communication.*

[Platt, 2005] J. Platt. Fastmap, MetricMap, and Landmark MDS are all Nyström algorithms. In Z. Ghahramani and R. Cowell, editors, *Proc. 10th International Conference on Artificial Intelligence and Statistics*, 2005.

[Press et al., 1992] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vettering. *Numerical recipes in C: the art of scientific computing.* Cambridge University Press, 2nd edition, 1992.

[Ross and Peköz, 2007] S.M. Ross and E.A. Peköz. *A Second Course in Probability.* www.ProbabilityBookstore.com, Boston, MA, 2007.

[Roweis and Saul, 2000] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(22):2323–2326, 2000.

[Schoenberg, 1935] I.J. Schoenberg. Remarks to maurice frechet's article *sur la définition axiomatique d'une classe d'espace distanciés vectoriellement applicable sur l'espace de hilbert. Annals of Mathematics*, 36:724–732, 1935.

[Schölkopf, 2001] B. Schölkopf. The kernel trick for distances. In T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 301–307. MIT Press, 2001.

[Schölkopf and Smola, 2002] B. Schölkopf and A. Smola. *Learning with Kernels.* MIT Press, 2002.

[Schölkopf et al., 1998] B. Schölkopf, A. Smola, and K-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[Shi and Malik, 2000] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[Spearman, 1904] C.E. Spearman. 'General intelligence' objectively determined and measured. *American Journal of Psychology*, 5:201–293, 1904.

[Steinwart, 2001] I. Steinwart. On the Influence of the Kernel on the Consistency of Support Vector Machines. *Journal of Machine Learning Research*, 37(2):726–755, 2001

[Stone, 1982] C.J. Stone. Optimal global rates of convergence for nonparametric regression. *Annals of Statistics*, 10(4):1040–1053, 1982.

[Tenenbaum, 1998] J.B. Tenenbaum. Mapping a manifold of perceptual observations. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.

[Tipping and Bishop, 1999A] M.E. Tipping and C.M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society*, 61(3):611, 1999A.

[Tipping and Bishop, 1999B] M.E. Tipping and C.M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999B.

[Tishby et al., 1999] N. Tishby, F.C. Pereira and William Bialek. The information bottleneck method. *Proc. of the 37th Annual Allerton Conference on Communication, Control and Computing*, 368-377, 1999.

[Viola and Jones, 2001] P. Viola and M. Jones. Robust real-time object detection. In *Second international workshop on statistical and computational theories of vision - modeling, learning, computing, and sampling*, 2001.

[Wilks, 1962] S. Wilks. *Mathematical Statistics*. John Wiley, 1962.

[Williams, 2001] C.K.I. Williams. On a Connection between Kernel PCA and Metric Multidimensional Scaling. In T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 675–681. MIT Press, 2001.

[Williams and Seeger, 2001] C.K.I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In Leen, Dietterich, and Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.