NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. (2010)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nla.743

Trace optimization and eigenproblems in dimension reduction methods

E. Kokiopoulou[1], J. Chen[2] and Y. Saad[2,*,†]

[1]*Seminar for Applied Mathematics, ETH, HG G J49, Rämistrasse 101, 8092 Zürich, Switzerland*
[2]*Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, U.S.A.*

SUMMARY

This paper gives an overview of the eigenvalue problems encountered in areas of data mining that are related to dimension reduction. Given some input high-dimensional data, the goal of dimension reduction is to map them to a low-dimensional space such that certain properties of the original data are preserved. Optimizing these properties among the reduced data can be typically posed as a trace optimization problem that leads to an eigenvalue problem. There is a rich variety of such problems and the goal of this paper is to unravel relationships between them as well as to discuss effective solution techniques. First, we make a distinction between projective methods that determine an explicit linear mapping from the high-dimensional space to the low-dimensional space, and nonlinear methods where the mapping between the two is nonlinear and implicit. Then, we show that all the eigenvalue problems solved in the context of explicit linear projections can be viewed as the projected analogues of the nonlinear or implicit projections. We also discuss kernels as a means of unifying linear and nonlinear methods and revisit some of the equivalences between methods established in this way. Finally, we provide some illustrative examples to showcase the behavior and the particular characteristics of the various dimension reduction techniques on real-world data sets. Copyright © 2010 John Wiley & Sons, Ltd.

Received 8 March 2010; Revised 29 June 2010; Accepted 2 July 2010

KEY WORDS: linear dimension reduction; nonlinear dimension reduction; principal component analysis; projection methods; locally linear embedding (LLE); Kernel methods; locality preserving projections (LPP); Laplacean eigenmaps

1. INTRODUCTION

The term 'data mining' refers to a broad discipline which includes such diverse areas as machine learning, data analysis, information retrieval, pattern recognition, and web-searching, to list just a few. The widespread use of linear algebra techniques in many subareas of data mining is remarkable. A prototypical area of data mining where numerical linear algebra techniques play a crucial role is that of *dimension reduction* which is the focus of this study. Dimension reduction is ubiquitous in applications ranging from pattern recognition and learning [1] to the unrelated fields of graph drawing [2, 3], materials research [4, 5], and magnetism [6].

The problem we have is to map some high-dimensional data to a low-dimensional space for various reasons, such as visualizing it, reducing the effect of noise or reducing computational

*Correspondence to: Y. Saad, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, U.S.A.
†E-mail: saad@cs.umn.edu

Contract/grant sponsor: NSF; contract/grant numbers: DMS-0810938, NSF-DMR 0940218
Contract/grant sponsor: Minnesota Supercomputer Institute

Copyright © 2010 John Wiley & Sons, Ltd.

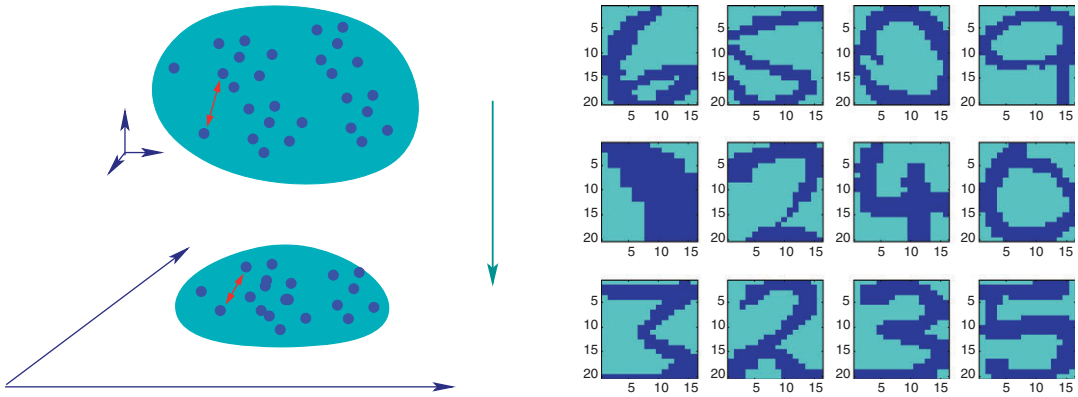

Figure 1. Left: Illustration of a certain mapping, in this trivial case from $\mathbb{R}^3$ to $\mathbb{R}^2$.
Right: a sample of 12-digit pictures [7].

cost when working with the data. Here, by mapping we mean that for each sample x_i from the high-dimensional space will find a low-dimensional version which we call y_i.

To be more specific, we are given a data matrix

$$X = [x_1, \ldots, x_n] \in \mathbb{R}^{m \times n} \tag{1}$$

for which we wish to find a low-dimensional analogue

$$Y = [y_1, \ldots, y_n] \in \mathbb{R}^{d \times n} \tag{2}$$

with $d \ll m$, which is a faithful representation of X in some sense. Formally, we are seeking a mapping:

$$\Psi : x \in \mathbb{R}^m \longrightarrow y = \Psi(x) \in \mathbb{R}^d.$$

Here x belongs to $\mathbb{R}^m$ where m can be in thousands or millions. Each of the n columns of the matrix X is a sample from this space, and n in turn can be in thousands or millions. The mapping Ψ is often not explicit. The only requirement is to be able to apply it to those data items in X, i.e. to columns of X. In other words, all we need is to *find a representative* y_i *in* $\mathbb{R}^d$ *for each sample* x_i. An illustration is shown on the left side of Figure 1. A question that is often asked is: *which of the two dimensions of* X *is typically larger?* The answer is that both cases occur and are important. The case when $n \leqslant m$ is called the *undersampled* case and will play an important role later in the discussion. One of the key issues in searching for a mapping Ψ is to specify what we mean by the requirement that Y be a *faithful representation* of X. In most cases, we will specify this by using a certain distance on X and on Y and by asking that 'closeness' be preserved with respect to this distance.

As an illustration, consider the problem of recognizing pictures of handwritten digits. This is an important problem and it provides a simple illustration to the basic ideas. Twelve sample pictures are shown on the right side of Figure 1. Each of these pictures is a 20×16 array of gray-level pixels[‡]. The 12 samples shown above are extracted (randomly) from a bigger data set containing actually $n = 390$ such pictures, which is publicly available [7]. This is a relatively small set of digit pictures compared with data sets arising in realistic situations. Whenever we deal with image data, it is common to 'vectorize' the arrays of pixels, i.e. to stack its columns (column-major order) or its rows (row-major order) into a long vector. In this case for each image we lexicographically list the gray-level data for each pixel, say row-major, hence we end up with a vector of length $m = 20 \times 16 = 320$ for each image. Therefore, we are in a situation where $n = 390$ and $m = 320$.

[‡]Actually for this particular data set, the gray level is either zero or one. In other situations it can take a larger number of values say from 0 to 255.

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

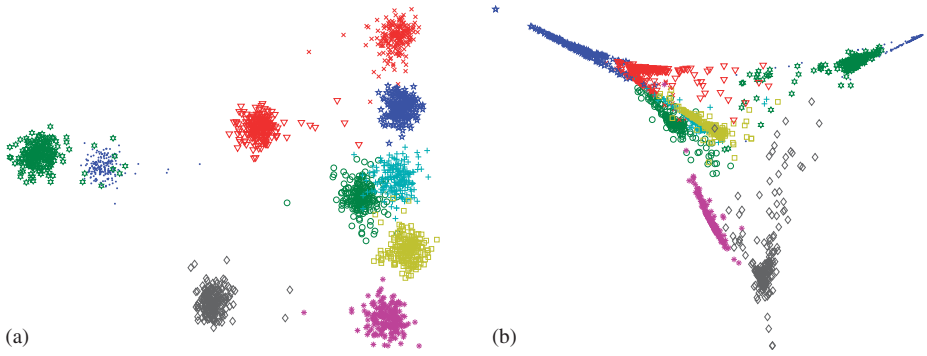

Figure 2. Two mappings of handwritten digits into 2-dimensional space: (a) LDA and (b) LLE.

A class of the dimension reduction methods in data mining consists of simply mapping directly the samples into a space of small dimension d. The smallest dimension is one, which is of limited interest. The case $d = 2$ is common for visualizing, for example, how the data samples are separated by the method under study. For instance, Figure 2 shows the results of two such methods, namely the classical Linear Discriminant Analysis method of Fisher, see, e.g. [8, p. 184], and the Locally Linear Embedding (LLE) of Roweis and Saul [9]. Details on these methods will be given later in the paper. Each different symbol (and color) corresponds to one of the digits from 0 to 9. There appears to be distinct blocks of data that form 'clusters' and there are 10 of them, clearly separated for LDA. Each of these clusters corresponds to one of the 10 digits.

This illustration will give us the opportunity to distinguish between two subclasses of methods used in data mining. There is a good reason why the projected data of the 390 digits onto 2D space appears to be better clustered on the left side of Figure 2 whereas those on the right side do not look as well separated. The reason is that the mapping Ψ, which projects an item from $\mathbb{R}^{390}$ to $\mathbb{R}^2$, exploits known information about the data. Specifically, we know to which digit each of the 390 images corresponds. We can *label* these with their digits and when we seek the mapping Ψ this information is utilized. This is referred to as *supervised learning*. One question that may arise is: why do this since we already have the information regarding the digits? In other words, there appears to be no reason to find a mapping to a lower dimension space if we have all the label information about these handwritten digits. One of the major goals of supervised learning is to use this information to derive a good mapping Ψ, which will then be applied not to the data X itself, but to *new data samples* which are not part of X, in an effort to find information about them. Suppose we are presented with a handwritten digit t that is *not* among the 390 samples. How can a machine recognize it? For human brains the task is fairly easy. In order to recognize the digit by computers, i.e. to find its label (a number from 0 to 9), we can use the data set X along with the label information available for it. For example we can project everything, i.e. the 390 digits and the test digit t, onto a 2-D plane as was done above and then find the closest, say, 8 items, among the 390 projected digits to the projection of t, using the Euclidean distance. We will then assign to t the most frequently occurring label among those of the 8 projected digits. This process is called k-nearest neighbor (k-NN) classification, as the labels of the k closest neighbors are used for determining the unknown class label of the test data sample t. See Figure 3 for an illustration for the case where $k = 8$. The data set X along with the labels is typically called *training set*. The digits are grouped into 10 groups (one for each digit) called *classes*, and there is a *label* associated with each class.

In contrast, *unsupervised clustering* is the task of finding subsets of the data such that items from the same subset are most similar and items from distinct subsets are most dissimilar. Here the degree of *similarity* can be measured by a simple distance (e.g. Euclidean) or via a *kernel*. Roughly speaking, the main idea of using kernels is to perform *implicitly* the learning task in a feature space of much higher dimension than the original space, with the hope that learning will be easier and more effective in such a feature space. The latter is obtained by applying a nonlinear

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

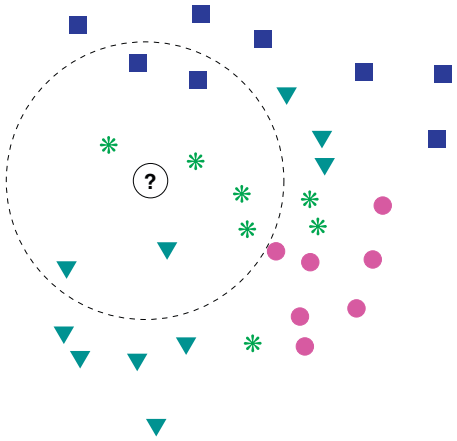

Figure 3. k-nearest neighbor classification.

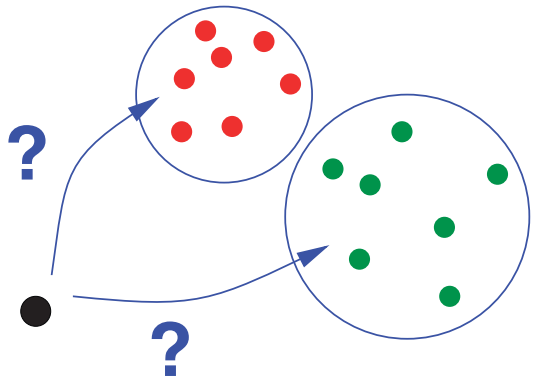

Figure 4. Illustration of the problem of classification (e.g. spam vs non-spam).

mapping Φ[§] to the original space to which columns of X belong. Then kernels enable the fast computation of inner products $\langle \Phi(x_i), \Phi(x_j) \rangle$, giving rise to a generalized notion of 'similarity' between any pair of items from X. In *unsupervised learning* no label information is available and no information is used other than the data itself. The LLE method mentioned earlier is in this category and its result on the handwritten digits, shown on the right of Figure 2, is rather impressive considering that only the pixels are used and that the mapping is from dimension 390 to dimension 2.

In *classification* (supervised learning), we are given a set of distinct data sets that are labeled (e.g. samples of handwritten digits labeled from 0 to 9), and when a new sample is presented to us we must determine to which of the sets it is most likely to belong. We have already seen the example of handwritten digits, where the problem is to recognize a digit given many labeled samples of already deciphered digits available in a given training data set. Another example is that of classifying e-mail messages into 'spam' and 'non-spam' (two classes). An illustration is shown on the left side of Figure 4.

In many data mining applications it is often the case that labeled data are hard or expensive to obtain, while at the same time there exists an abundance of unlabeled data. For example, image annotation is a time-consuming and laborious task that must be performed by human experts, whereas unlabeled/un-annotated images are very easy to collect. This scenario of limited

[§]This mapping is not to be confused with Ψ.

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

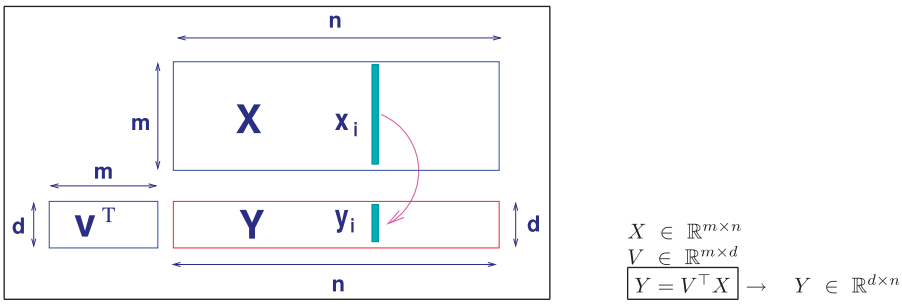

Figure 5. Linear projection.

supervision information is termed *semi-supervised* and it presents a challenge to dimension reduction methods that must exploit both labeled and unlabeled data effectively.

In order to perform these tasks, whether in a (semi-)supervised or unsupervised setting, it is common to first process the given data sets (e.g. the set of handwritten digits) in order to reduce its dimension, i.e. to find a data set of much lower dimension than the original one, but which preserves its main features. What is often misunderstood is that this dimension reduction is not done for the sole purpose of reducing cost, but mainly for reducing the effect of noise and extracting the main features of the data. For this reason, the low-dimensional vectors y_i are also known as *features* and the dimension reduction process is sometimes referred to as *feature extraction*.

There have been two classes of methods proposed for dimension reduction. The first class of methods can be termed *linear* or *projective*. This includes all methods whereby the data matrix X is explicitly transformed into a low-dimensional version Y by a linear transformation; that is, the mapping Ψ corresponds to a linear transformation in this case. Then these projective methods find an $m \times d$ $(m \gg d)$ matrix V and express the reduced dimension data as $Y = V^{\mathrm{T}}X$. Figure 5 summarizes the notation and illustrates the linear projection process. The methods in the second class, called *nonlinear* methods, do not rely on explicit projections and find *directly* the low-dimensional data matrix Y. In this case, the mapping Ψ is implicit and inherently nonlinear [10]. Both types of dimension reduction methods can be extended to supervised versions, where the class labels are taken into account when performing the reduction step.

We have already mentioned that Y is sought such that it is a faithful representation of X, i.e. certain properties of X are preserved in the reduced space. Examples of properties to be preserved may include the global geometry, neighborhood information such as local neighborhoods [9, 11] and local tangent space [12], distances between data samples [13, 14] or angles formed by adjacent line segments [15].

The goal of this paper is (i) to highlight the use of eigenproblems in dimension reduction as well as provide an exposition of a few relevant techniques and (ii) to unravel some of the relationships between these dimension reduction methods, their supervised counterparts, and the optimization problems they rely upon. Although the paper includes an overview of some relevant techniques, it is not meant to be exhaustive, as the main goal is to provide a unified view of such methods and reveal the connections between them. In addition, the paper will not describe the details of the various applications. Instead these applications will be summarized and expressed in simple mathematical terms with the goal of showing the objective function that is optimized in each case. In addition, two main observations will be made in this paper. The first is about a distinction between the projective methods and the nonlinear ones. Specifically, the eigenvalue problem solved in the linear case consists of applying a projection technique, i.e. a Rayleigh–Ritz projection method, as it leads to the solution of an eigenvalue problem in the space spanned by the columns of the data matrix X^{T}. The second is that these two families of methods can be brought together thanks to the use of kernels. These observations will strengthen a few similar observations made in a few earlier papers, e.g. [16–18].

The remainder of this paper is organized as follows. Section 2 summarizes a few well-known results of linear algebra that will be exploited repeatedly in the paper. Then, Sections 3 and 4

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

provide a brief overview of the nonlinear and linear methods respectively for dimension reduction. Section 5 discusses dimension reduction in supervised settings, where the class labels of the data are taken into account, and Section 6 deals with the semi-supervised scenario, where not all the training samples are associated with a class label. Section 7 provides an analysis of the relationships between the different methods as well as connections to methods from different areas, such as spectral clustering and projection techniques for eigenvalue problems. Kernelized versions of different linear dimension reduction methods are discussed in Section 8, along with various relationships with their nonlinear counterparts. Section 9 provides illustrative examples for data visualization and classification of handwritten digits and faces. Finally, Section 10 briefly mentions some numerical techniques beyond trace optimizations for dimension reduction, and the paper ends with a conclusion in Section 11.

2. PRELIMINARIES

First, given a symmetric matrix A of dimension $n \times n$ and an arbitrary orthogonal matrix V of dimension $n \times d$, then the trace of $V^{\mathrm{T}}AV$ is maximized when V is an orthogonal basis of the eigenspace associated with the (algebraically) largest eigenvalues. In particular, it is achieved for the eigenbasis itself: if eigenvalues are labeled in a decreasing order and $u_1, \ldots, u_d$ are eigenvectors associated with the first d eigenvalues $\lambda_1, \ldots, \lambda_d$, and $U = [u_1, \ldots, u_d]$, with $U^{\mathrm{T}}U = I$, then,

$$\max_{\substack{V \in \mathbb{R}^{n \times d} \\ V^{\mathrm{T}}V = I.}} \mathrm{Tr}[V^{\mathrm{T}}AV] = \mathrm{Tr}[U^{\mathrm{T}}AU] = \lambda_1 + \cdots + \lambda_d, \tag{3}$$

While this result is seldom explicitly stated on its own in standard textbooks, it is an immediate consequence of the Courant–Fisher characterization, see, e.g. [19, 20]. It is important to note that the optimal V is far from being unique. In fact, any V which is an orthonormal basis of the eigenspace associated with the first d eigenvalues will be optimal. In other words, what matters is the subspace rather than a particular orthonormal basis for it.

The key point is that to maximize the trace in (3), one needs to solve a standard eigenvalue problem. In many instances, we need to maximize $\mathrm{Tr}[V^{\mathrm{T}}AV]$ subject to a new normalization constraint for V, one that requires that V to be B-orthogonal, i.e. $V^{\mathrm{T}}BV = I$. Assuming that A is symmetric and B positive definite, we know that there are n real eigenvalues for the generalized problem $Au = \lambda Bu$, with B-orthogonal eigenvectors. If these eigenvalues are labeled in a decreasing order, and if $U = [u_1, \ldots, u_d]$ is the set of eigenvectors associated with the first d eigenvalues, with $U^{\mathrm{T}}BU = I$, then we have

$$\max_{\substack{V \in \mathbb{R}^{n \times d} \\ V^{\mathrm{T}}BV = I.}} \mathrm{Tr}[V^{\mathrm{T}}AV] = \mathrm{Tr}[U^{\mathrm{T}}AU] = \lambda_1 + \cdots + \lambda_d, \tag{4}$$

In reality, Problem (4) often arises as a simplification of an objective function that is more difficult to maximize, namely:

$$\max_{\substack{V \in \mathbb{R}^{n \times d} \\ V^{\mathrm{T}}CV = I.}} \frac{\mathrm{Tr}[V^{\mathrm{T}}AV]}{\mathrm{Tr}[V^{\mathrm{T}}BV]}, \tag{5}$$

Here B and C are assumed to be symmetric and positive definite for simplicity. The matrix C defines the desired orthogonality and in the simplest case it is just the identity matrix. The original version shown above has resurfaced in the recent years, see, e.g. [21–25] among others. Although we will not give the above problem as much attention as the more standard problem (4), it is important to give an idea about the way it is commonly solved. There is no loss of generality in assuming that C is the identity. As B is assumed to be positive definite[¶], it is not difficult to see

[¶]We can relax the assumptions: B can be positive semi-definite, but for the problem to be well-posed its null space must be of dimension less than d. Also if A is positive semi-definite, we must assume that $\mathrm{Null}(A) \cap \mathrm{Null}(B) = \emptyset$.

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

that there is a maximum μ that is reached for a certain (non-unique) orthogonal matrix, which we will denote by U. Then, $\mathrm{Tr}[V^{\mathrm{T}}AV]-\mu\mathrm{Tr}[V^{\mathrm{T}}BV]\leqslant 0$ for any orthogonal V. This means that for this μ we have $\mathrm{Tr}[V^{\mathrm{T}}(A-\mu B)V]\leqslant 0$ for any orthogonal V, and also $\mathrm{Tr}[U^{\mathrm{T}}(A-\mu B)U]=0$. Therefore, we have the following necessary condition for the pair μ, U to be optimal:

$$\max_{V^{\mathrm{T}}V=I} \mathrm{Tr}[V^{\mathrm{T}}(A-\mu B)V]=\mathrm{Tr}[U^{\mathrm{T}}(A-\mu B)U]=0. \tag{6}$$

According to (3), the maximum trace of $V^{\mathrm{T}}(A-\mu B)V$ is simply the sum of the largest d eigenvalues of $A-\mu B$ and U is the set of corresponding eigenvectors. If μ maximizes the trace ratio (5) (with $C=I$), then the sum of the largest d eigenvalues of the pencil $A-\mu B$ equals zero, and the corresponding eigenvectors form the desired optimal solution of (5).

When B is positive definite, it can be seen that the function

$$f(\theta)=\max_{V^{\mathrm{T}}V=I} \mathrm{Tr}[V^{\mathrm{T}}(A-\theta B)V]$$

is a decreasing function of θ. For $\theta=0$ we have $f(\theta)>0$. For $\theta>\lambda_{\max}(A,B)$ we have $f(\theta)<0$, where $\lambda_{\max}(A,B)$ is the largest generalized eigenvalue of the pencil (A,B). Finding the optimal solution will involve a search for the (unique) root of $f(\theta)$. In [21, 22, 25] algorithms were proposed to solve (5) by computing this root and by exploiting the above relationships. It appears clearly that it will be more expensive to solve (5) than (4), because the search for the root μ will typically involve solving several eigenvalue problems instead of just one. However, this difference can be mitigated when these eigenvalue problems are solved only approximately.

3. NONLINEAR DIMENSION REDUCTION

We start with an overview of the nonlinear methods. In what follows, we discuss LLE and Laplacean Eigenmaps, which are the most representative nonlinear methods for dimension reduction. These methods begin with the construction of a weighted graph, which captures some information about the local neighborhood structure of the data. In the sequel, we refer to this graph as the *affinity graph*. Specifically, the affinity (or adjacency) graph is a graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$ whose nodes, which belong to the set $\mathcal{V}$, represent the data samples. The edges of this graph can be defined for example by taking a certain nearness measure and including all samples within a radius ε of a given node, to its adjacency list. Alternatively, one can include those k nodes that are the nearest neighbors to x_i. In the latter case it is called the k-NN graph. It is typical to assign weights w_{ij} on the edges $e_{ij}\in\mathcal{E}$ of the affinity graph. Note that the weights can either be symmetric or asymmetric, and different assignments will be clear in the exposition of specific dimension reduction methods. The affinity graph along with these weights then defines a matrix W whose entries are the weights w_{ij} that are non-zero only for adjacent nodes in the graph.

3.1. LLE

In *Locally Linear Embedding* (LLE), the construction of the affinity graph is based on the assumption that the samples lie on some high-dimensional manifold, hence each sample is approximately expressed as a linear combination of a few neighbors, see [9, 26]. Thus, the affinity matrix is built by computing optimal weights which will relate a given sample to its neighbors in some locally optimal way. The reconstruction error for sample i can be measured by

$$\left\| x_i-\sum_j w_{ij}x_j \right\|_2^2. \tag{7}$$

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

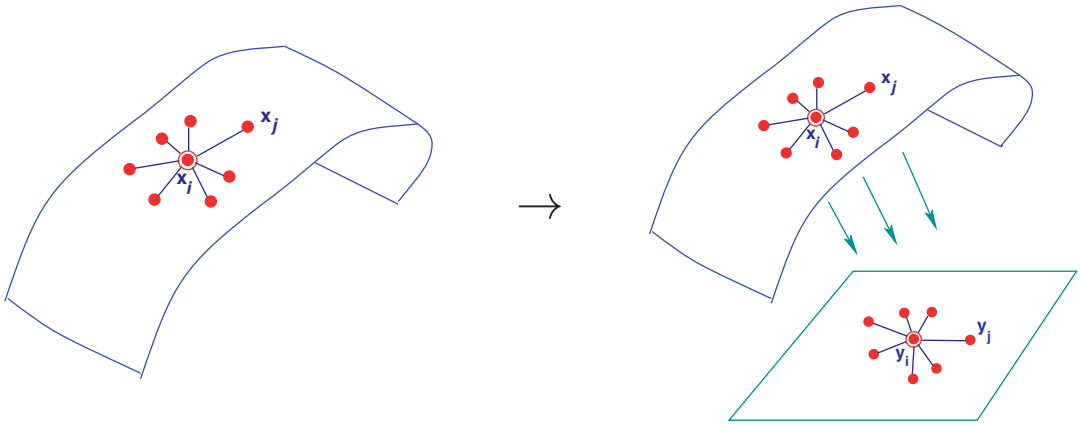

Figure 6. Left: Local neighborhood and construction of the weight matrix in LLE. Right: LLE finds a low-dimensional space (Y) that best reproduces the local neighborhoods of the original space.

The weights w_{ij} represent the linear coefficients for (approximately) reconstructing the sample x_i from its neighbors $\{x_j\}$, with $w_{ij}=0$ if x_j is not one of the k nearest neighbors of x_i. We can set $w_{ii}\equiv 0$, for all i. The coefficients are scaled so that their sum is unity, i.e.

$$\sum_j w_{ij}=1. \tag{8}$$

Determining the w_{ij}'s for a given sample x_i is a *local* calculation, in the sense that it only involves x_i and its nearest neighbors. As a result, computing the weights will be fairly inexpensive; an explicit solution can be extracted by solving a small linear system which involves a 'local' Grammian matrix; for details see [9, 26]. After this phase is completed we have a matrix W such that each column x_i of the data set is well represented by a linear combination $\sum_j w_{ij}x_j$. In other words, $X\approx XW^{\mathrm{T}}$, i.e. X^{T} is a set of approximate left null vectors of $I-W$.

The procedure then seeks d-dimensional vectors y_i, $i=1,\ldots,n$ so that the same relationship is satisfied between the matrix W and the y_i's. An illustration is shown in Figure 6. This is achieved by minimizing the objective function

$$\mathscr{F}_{\mathrm{LLE}}(Y)=\sum_i \left\| y_i-\sum_j w_{ij}y_j \right\|_2^2. \tag{9}$$

LLE imposes two constraints to this optimization problem: (i) the mapped coordinates must be centered at the origin and (ii) the embedded vectors must have unit covariance:

$$\sum_i y_i=0 \quad \text{and} \quad \frac{1}{n}\sum_i y_i y_i^{\mathrm{T}}=I. \tag{10}$$

The objective function (9) is minimized with these constraints on Y.

We can rewrite (9) as a trace by noting that $\mathscr{F}_{\mathrm{LLE}}(Y)=\|Y-YW^{\mathrm{T}}\|_F^2$, and this leads to:

$$\mathscr{F}_{\mathrm{LLE}}(Y)=\mathrm{Tr}[Y(I-W^{\mathrm{T}})(I-W)Y^{\mathrm{T}}]. \tag{11}$$

Therefore the new optimization problem to solve is**

$$\min_{\substack{Y\in\mathbb{R}^{d\times n}\\ YY^{\mathrm{T}}=I.}} \mathrm{Tr}[Y(I-W^{\mathrm{T}})(I-W)Y^{\mathrm{T}}], \tag{12}$$

**The final y_i's are obtained by translating and scaling each column of Y.

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

The solution of the problem is obtained from the set of eigenvectors associated with the d smallest eigenvalues of $M \equiv (I - W^{\mathrm{T}})(I - W)$:

$$(I - W^{\mathrm{T}})(I - W)u_i = \lambda_i u_i, \quad Y = [u_2, \ldots, u_{d+1}]^{\mathrm{T}}. \tag{13}$$

Note that the eigenvector associated with the eigenvalue zero is discarded and that matrix Y is simply the set of bottom eigenvectors of $(I - W^{\mathrm{T}})(I - W)$ associated with the 2nd to $(d+1)$-th eigenvalues. We will often refer to the matrix $M = (I - W^{\mathrm{T}})(I - W)$ as the LLE matrix.

3.2. Laplacean Eigenmaps

The *Laplacean Eigenmaps* technique [11, 27] is rather similar to LLE. It uses different weights to represent locality and a slightly different objective function. Two common choices are weights of the heat (or Gaussian) kernel $w_{ij} = \exp(-\|x_i - x_j\|_2^2/\sigma^2)$ or constant weights ($w_{ij} = 1$ if i and j are adjacent, $w_{ij} = 0$ otherwise). The first choice of weights is very popular and it differs from the second choice mainly in that the neighbors of a data sample are treated in a non-uniform way. In particular, the role of the parameter σ is to assign a relatively large (resp. small) weight to the closest (resp. farthest) neighbors. In this way the closest neighbors are given more importance. The choice of the parameter σ is crucial for the performance of methods that use the Gaussian kernel.

Once this graph is available, a Laplacean matrix of the graph is constructed by setting a diagonal matrix D with diagonal entries $d_{ii} = \sum_j w_{ij}$. The matrix

$$L \equiv D - W$$

is the *Laplacean* of the weighted graph defined above. Note that the row-sums of the matrix L are zero by the definition of D, so $L\mathbf{1} = 0$, and therefore L is singular. The problem in Laplacean Eigenmaps is then to minimize

$$\mathscr{F}_{\mathrm{EM}}(Y) = \sum_{i,j=1}^{n} w_{ij} \|y_i - y_j\|_2^2 \tag{14}$$

subject to an orthogonality constraint that uses the matrix D for scaling:

$$Y D Y^{\mathrm{T}} = I.$$

The rationale for this approach is to *put a penalty for mapping nearest neighbor nodes in the original graph to distant samples in the low-dimensional data*.

Compare (14) and (9). The difference between the two is subtle and one might ask if (14) can also be converted into a trace optimization problem similar to (12). As it turns out $\mathscr{F}_{\mathrm{EM}}$ can be written as a trace and this brings the method quite close to LLE in spirit. This is because it can be easily shown that [28, 29]:

$$\mathscr{F}_{\mathrm{EM}}(Y) = 2\mathrm{Tr}[Y(D - W)Y^{\mathrm{T}}]. \tag{15}$$

Therefore, the new optimization problem to solve is

$$\min_{\substack{Y \in \mathbb{R}^{d \times n} \\ Y D Y^{\mathrm{T}} = I.}} \mathrm{Tr}[Y(D - W)Y^{\mathrm{T}}], \tag{16}$$

The solution Y to this optimization problem can be obtained from the eigenvectors associated with the d smallest eigenvalues of the generalized eigenvalue problem

$$(D - W)u_i = \lambda_i D u_i, \quad Y = [u_2, \ldots, u_{d+1}]^{\mathrm{T}}. \tag{17}$$

One can also solve a standard eigenvalue problem by making a small change in variables, and this is useful to better see links with other methods. Indeed, it would be useful to standardize the

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

constraint YDY^{T} so that the diagonal scaling does not appear. For this we set $\hat{Y} = YD^{1/2}$ and $\hat{W} = D^{-1/2}WD^{-1/2}$, and this simplifies (16) into:

$$\min_{\substack{\hat{Y}\in\mathbb{R}^{d\times n}\\ \hat{Y}\hat{Y}^{\mathrm{T}}=I.}} \mathrm{Tr}[\hat{Y}(I-\hat{W})\hat{Y}^{\mathrm{T}}]. \tag{18}$$

In this case, (17) yields:

$$(I-\hat{W})\hat{u}_i = \lambda_i \hat{u}_i, \quad Y = [\hat{u}_2, \ldots, \hat{u}_{d+1}]^T D^{1/2}. \tag{19}$$

The matrix $\hat{L} = I - \hat{W} = D^{-1/2}LD^{-1/2}$ is called the *normalized Laplacean*.

4. LINEAR DIMENSION REDUCTION

The methods in the previous section do not provide an explicit function that maps a vector x into its low-dimensional representation y in d-dimensional space. This mapping is only known for each of the vectors x_i of the data set X, i.e. we know how to associate a low-dimensional item y_i to each sample x_i. In some applications it is important to be able to find the mapping y for an arbitrary, 'out-of-sample' vector x. The methods discussed in this section have been developed in part to address this issue. They are based on an explicit (linear) mapping defined by a matrix $V \in \mathbb{R}^{m\times d}$. These projective techniques replace the original data X by a matrix of the form

$$Y = V^{\mathrm{T}}X \quad \text{where } V \in \mathbb{R}^{m\times d}. \tag{20}$$

Once the matrix V has been 'learned', i.e. extracted, each vector x_i can be projected to the reduced space by simply computing $y_i = V^{\mathrm{T}}x_i$. If V is an orthogonal matrix, then Y represents the orthogonal projection of X into the V-space.

4.1. PCA

The best known technique in this category is *Principal Component Analysis* (PCA) [30]. PCA computes an orthonormal matrix V so that the variance of the projected vectors is maximized, i.e, V is the maximizer of

$$\max_{\substack{V\in\mathbb{R}^{m\times d}\\ V^{\mathrm{T}}V=I.}} \sum_{i=1}^{n} \left\| y_i - \frac{1}{n}\sum_{j=1}^{n} y_j \right\|_2^2, \quad y_i = V^{\mathrm{T}}x_i, \tag{21}$$

Recalling that $\mathbf{1}$ denotes the vector of all ones, the objective function in (21) becomes

$$\mathscr{F}_{\mathrm{PCA}}(Y) = \sum_{i=1}^{n} \left\| y_i - \frac{1}{n}\sum_{j=1}^{n} y_j \right\|_2^2 = \mathrm{Tr}\left[V^{\mathrm{T}}X\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}\right)X^{\mathrm{T}}V \right].$$

In the end, the above optimization can be restated as

$$\max_{\substack{V\in\mathbb{R}^{m\times d}\\ V^{\mathrm{T}}V=I.}} \mathrm{Tr}\left[V^{\mathrm{T}}X\left(I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}\right)X^{\mathrm{T}}V \right], \tag{22}$$

In the sequel we will denote by $\bar{X}$ the matrix $X(I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}})$, which is simply the matrix with centered data, i.e. each column is $\bar{x}_i = x_i - \mu$, where μ is the mean of X, $\mu = \sum x_i/n$. As the matrix in (22) can be written as $V^{\mathrm{T}}\bar{X}\bar{X}^{\mathrm{T}}V$, (22) becomes

$$\max_{\substack{V\in\mathbb{R}^{m\times d}\\ V^{\mathrm{T}}V=I.}} \mathrm{Tr}[V^{\mathrm{T}}\bar{X}\bar{X}^{\mathrm{T}}V], \tag{23}$$

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

The orthogonal matrix V which maximizes the trace in (23) is simply the set of left singular vectors of $\bar{X}$ associated with the largest d singular values,

$$[\bar{X}\bar{X}^{\mathrm{T}}]v_i = \lambda_i v_i. \tag{24}$$

The matrix $V = [v_1, \ldots, v_d]$ is used for projecting the data, hence $Y = V^{\mathrm{T}}\bar{X}$. If $\bar{X} = U\Sigma Z^{\mathrm{T}}$ is the SVD of $\bar{X}$, the solution to the above optimization problem is $V = U_d$, the matrix of the first d left singular vectors of X, hence, denoting by Σ_d the top left $d \times d$ block of Σ, and Z_d the matrix of the first d columns of Z, we obtain

$$Y = U_d^{\mathrm{T}}\bar{X} = \Sigma_d Z_d^{\mathrm{T}}. \tag{25}$$

As it turns out, maximizing the variance on the projected space is equivalent to minimizing the projection error

$$\|\bar{X} - VV^{\mathrm{T}}\bar{X}\|_F^2 = \|\bar{X} - VY\|_F^2.$$

This is because a little calculation will show that

$$\|\bar{X} - VY\|_F^2 = \mathrm{Tr}[(\bar{X} - VY)^{\mathrm{T}}(\bar{X} - VY)] = \mathrm{Tr}[\bar{X}^{\mathrm{T}}\bar{X}] - \mathrm{Tr}[V^{\mathrm{T}}\bar{X}\bar{X}^{\mathrm{T}}V].$$

The matrix VV^{T} is an orthogonal projector onto the span of V. The samples $Vy_i \in \mathbb{R}^m$ are sometimes referred to as *reconstructed points*. PCA minimizes the sum of the squares of the distance between any sample in the data set and its reconstruction, i.e. its projection.

4.2. MDS and ISOMAP[††]

In metric *Multi-Dimensional Scaling* (metric MDS) the problem posed is to project data in such a way that distances $\|y_i - y_j\|_2$ between projected samples are closest to the original distances $\|x_i - x_j\|_2$. Instead of solving the problem in this form, MDS uses a criterion based on inner products.

It is now assumed that the data are centered at zero hence we replace X by $\bar{X}$. An important result used is that one can recover distances from inner products and vice versa. The matrix of inner products, i.e. the Grammian of $\bar{X}$, defined by

$$G = [\langle \bar{x}_i, \bar{x}_j \rangle]_{i,j=1,\ldots,n} \tag{26}$$

determines completely the distances, since $\|\bar{x}_i - \bar{x}_j\|^2 = g_{ii} + g_{jj} - 2g_{ij}$. The reverse can also be done, i.e. one can determine the inner products from distances by 'inverting' the above relationships. Indeed, under the assumption that the data are centered at zero, it can be shown that [31]

$$g_{ij} = \frac{1}{2}\left[\frac{1}{n}\sum_k (s_{ik} + s_{jk}) - s_{ij} - \frac{1}{n^2}\sum_{k,l} s_{kl}\right],$$

where $s_{ij} = \|\bar{x}_i - \bar{x}_j\|^2$. In matrix form, the relationship is:

$$G = -\frac{1}{2}\left[I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}\right] S \left[I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}\right], \quad S = [s_{ij}]_{i,j=1,\ldots,n}.$$

As a result of the above equality, in order to find a d-dimensional projection which preserves pairwise distances as possible, we need to find a $d \times n$ matrix Y whose Grammian $Y^{\mathrm{T}}Y$ is close to G, the Grammian of X, i.e. we need to find the solution of

$$\min_{Y \in \mathbb{R}^{d \times n}} \|G - Y^{\mathrm{T}}Y\|_F^2. \tag{27}$$

[††]ISOMAP is essentially a nonlinear method, which is presented here only because it is closely related to PCA.

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

Let $G = Z\Lambda Z^{\mathrm{T}}$ be the eigenvalue decomposition of G, where it is assumed that the eigenvalues are labeled from largest to smallest. Then the solution to (27) is $Y = \Lambda_d^{1/2} Z_d^{\mathrm{T}}$ where Z_d consists of the first d columns of Z, Λ_d is the $d \times d$ upper left block of Λ. Note that with respect to the SVD of $\bar{X}$ this is equal to $\Sigma_d Z_d^{\mathrm{T}}$, which is identical to the result obtained with PCA; see Equation (25). *Thus metric MDS gives the same exact result as PCA*. However, it arrives at this result using a different path. PCA uses the covariance matrix, whereas MDS uses the Gram matrix. From a computational cost point of view, there is no real difference if the calculation is based on the SVD of $\bar{X}$. We should note that the solution to (27) is unique only up to orthogonal transformations. This is because a transformation such as $\hat{Y} = QY$ of Y, where Q is orthogonal, will not change distances between y-samples.

Finally, we mention in passing that the technique of ISOMAP [13] essentially performs the same steps as MDS, except that the Grammian $G = \bar{X}^{\mathrm{T}}\bar{X}$ is replaced by a pseudo-Grammian $\hat{G}$ obtained from *geodesic distances* between the samples x_i:

$$\hat{G} = -\frac{1}{2}\left[I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}\right]\hat{S}\left[I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}\right], \quad \hat{S} = [\hat{s}_{ij}]_{i,j=1,\ldots,n},$$

where $\hat{s}_{ij}$ is the squared shortest graph distance between x_i and x_j.

4.3. LPP

The *Locality Preserving Projections* (LPP) [28] is a graph-based projective technique. It projects the data so as to preserve a certain *affinity graph* constructed from the data. LPP defines the projected samples in the form $y_i = V^{\mathrm{T}} x_i$ by *putting a penalty for mapping nearest neighbor nodes in the original graph to distant samples in the projected data*. Therefore, the objective function to be minimized is identical with that of Laplacean Eigenmaps,

$$\mathscr{F}_{\mathrm{LPP}}(Y) = \sum_{i,j=1}^{n} w_{ij} \|y_i - y_j\|_2^2.$$

The matrix V, which is the actual unknown, is implicitly represented in the above function through the dependence of the y_i's on V. Writing $Y = V^{\mathrm{T}}X$, we reach the optimization problem,

$$\min_{\substack{V \in \mathbb{R}^{m \times d} \\ V^{\mathrm{T}}(XDX^{\mathrm{T}})V = I.}} \mathrm{Tr}[V^{\mathrm{T}}X(D-W)X^{\mathrm{T}}V], \tag{28}$$

whose solution can be computed from the generalized eigenvalue problem

$$X(D-W)X^{\mathrm{T}}v_i = \lambda_i XDX^{\mathrm{T}}v_i. \tag{29}$$

Similar to Eigenmaps, the smallest d eigenvalues and eigenvectors must be computed.

It is simpler to deal with the 'normalized' case of LPP, by scaling the set Y as in the case of Laplacean Eigenmaps (see Equation (18)). We define $\hat{Y} = YD^{1/2} = V^{\mathrm{T}}XD^{1/2}$. Thus, if $\hat{X} = XD^{1/2}$, we have $\hat{Y} = V^{\mathrm{T}}\hat{X}$, and the above problem then becomes

$$\min_{\substack{V \in \mathbb{R}^{m \times d} \\ V^{\mathrm{T}}(\hat{X}\hat{X}^{\mathrm{T}})V = I.}} \mathrm{Tr}[V^{\mathrm{T}}\hat{X}(I-\hat{W})\hat{X}^{\mathrm{T}}V], \tag{30}$$

where $\hat{W}$ is the same matrix as in (18). The eigenvalue problem to solve is now

$$\hat{X}(I-\hat{W})\hat{X}^{\mathrm{T}}v_i = \lambda_i \hat{X}\hat{X}^{\mathrm{T}}v_i. \tag{31}$$

The projected data y_i is defined by $y_i = V^{\mathrm{T}}x_i$ for each i, where $V = [v_1, \ldots, v_d]$.

4.4. ONPP

Orthogonal Neighborhood Preserving Projection (ONPP) [16, 29] seeks an orthogonal mapping of a given data set so as to best preserve the same affinity graph as LLE. In other words, ONPP

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

is an orthogonal projection version of LLE. The projection matrix V in ONPP is determined by minimizing the same objective function as in (11), with the additional constraint that Y is of the form $Y = V^{\mathrm{T}}X$ and the columns of V be orthonormal, i.e. $V^{\mathrm{T}}V = I$. The optimization problem becomes

$$\min_{\substack{V \in \mathbb{R}^{m\times d} \\ V^{\mathrm{T}}V = I.}} \mathrm{Tr}[V^{\mathrm{T}}X(I - W^{\mathrm{T}})(I - W)X^{\mathrm{T}}V]. \tag{32}$$

Its solution is the basis of the eigenvectors associated with the d smallest eigenvalues of the matrix $\tilde{M} \equiv X(I - W^{\mathrm{T}})(I - W)X^{\mathrm{T}} = XMX^{\mathrm{T}}$.

$$X(I - W^{\mathrm{T}})(I - W)X^{\mathrm{T}}u_i = \lambda u_i. \tag{33}$$

Then the projector V is $[u_1, u_2, \ldots, u_d]$ and results in the projected data $Y = V^{\mathrm{T}}X$.

The assumptions that were made when defining the weights w_{ij} in Section 3.1 imply that the $n \times n$ matrix $I - W$ is singular due to Equation (8). In the case when $m > n$ the matrix $\tilde{M}$, which is of size $m \times m$, is at most of rank n and it is therefore singular. In the case when $m \leqslant n$, $\tilde{M}$ is not necessarily singular but it is observed in practice that ignoring the smallest eigenvalue is helpful [29].

4.5. Other variations on the locality preserving theme

A few possible variations of the methods discussed above can be developed. As was seen, ONPP is one such variation which adapts the LLE affinity graph and seeks a projected data which preserves this graph just as in LLE. Another very simple option is to solve the same optimization problem as ONPP but require the same orthogonality of the projected data as LLE, namely: $YY^{\mathrm{T}} = I$. This yields the constraint $V^{\mathrm{T}}XX^{\mathrm{T}}V = I$ instead of the $V^{\mathrm{T}}V = I$ required in ONPP. In [16] we called this *Neighborhood Preserving Projections* (NPP). The resulting new optimization problem is the following modification of (32)

$$\min_{\substack{V \in \mathbb{R}^{m\times d} \\ V^{\mathrm{T}}XX^{\mathrm{T}}V = I.}} \mathrm{Tr}[V^{\mathrm{T}}X(I - W^{\mathrm{T}})(I - W)X^{\mathrm{T}}V], \tag{34}$$

and the new solution is

$$X(I - W^{\mathrm{T}})(I - W)X^{\mathrm{T}}u_i = \lambda(XX^{\mathrm{T}})u_i. \tag{35}$$

As before, $V = [u_1, \ldots, u_d]$ and $y_i = V^{\mathrm{T}}x_i$, $i = 1, \ldots, n$.

Another variation goes in the other direction by using the objective function of LPP (using graph Laplaceans) and requiring the data to be orthogonally projected:

$$\min_{\substack{V \in \mathbb{R}^{m\times d} \\ V^{\mathrm{T}}V = I.}} \mathrm{Tr}[V^{\mathrm{T}}X(D - W)X^{\mathrm{T}}V], \tag{36}$$

This was referred to as *Orthogonal Locality Preserving Projections* (OLPP) in [16]. Note in passing that a different technique was developed in [32] and named Orthogonal Laplacean faces, which is also sometimes referred to as OLPP. We will not refer to this method in this paper and there is therefore no confusion.

5. SUPERVISED DIMENSION REDUCTION

We have already mentioned that supervised methods, unlike unsupervised methods, take into account class labels during dimension reduction. It has been observed in general that supervised methods for dimension reduction perform better than unsupervised methods in many classification tasks. In what follows, we first describe supervised versions of the above graph-based methods and then we discuss Linear Discriminant Analysis (LDA), which is one of the most popular supervised techniques for linear dimension reduction.

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

5.1. Supervised graph-based methods

As discussed so far, the methods in Section 4 do not make use of class labels. It is possible to develop supervised versions of those methods by taking the class labels into account. Assume that we have c classes and that the data are organized, without loss of generality, as $X_1, \ldots, X_c$ with $X_i \in \mathbb{R}^{m \times n_i}$, where n_i denotes the number of samples that belong to the ith class. In other words, assume that the data samples are ordered according to their class membership.

In supervised methods the class labels are used to build the graph. The main idea is to build the graph in a discriminant way in order to reflect the categorization of the data into different classes. One simple approach is to impose that an edge $e_{ij} = (x_i, x_j)$ exists if and only if x_i and x_j belong to the same class. In other words, we make adjacent those nodes that belong to the same class. For instance, preserving localities in such a supervised graph, will result in samples from the same class being projected close-by in the reduced space.

Consider now the structure of the induced adjacency matrix H. Observe that the data graph $\mathcal{G}$ consists of c cliques, since the adjacency relationship between two nodes reflects their class membership. Let $\mathbf{1}_{n_j}$ denote the vector of all ones, with length n_j, and $H_j = \frac{1}{n_j}\mathbf{1}_{n_j}\mathbf{1}_{n_j}^{\mathrm{T}} \in \mathbb{R}^{n_j \times n_j}$ be the block corresponding to the jth class. The $n \times n$ adjacency matrix H will be of the following form

$$H = \mathrm{diag}[H_1, H_2, \ldots, H_c]. \tag{37}$$

Thus, the (1,1) diagonal block is of size $n_1 \times n_1$ and has the constant entries $1/n_1$, the (2,2) diagonal block is of size $n_2 \times n_2$ and has the constant entries $1/n_2$, and so on. Using the above supervised graph in the graph-based dimension reduction methods yields their supervised versions.

5.2. LDA

The principle used in *Linear Discriminant Analysis* (LDA) is to project the original data linearly in such a way that the low-dimensional data is best separated. Fisher's Linear Discriminant Analysis, see, e.g. Webb [1], seeks to project the data in low-dimensional space so as to maximize the ratio of the 'between scatter' measure over 'within scatter' measure of the classes, which are defined next. Let μ be the mean of all the data sets, and $\mu^{(k)}$ be the mean of the k-th class, which is of size n_k, and define the two matrices

$$S_B = \sum_{k=1}^{c} n_k(\mu^{(k)} - \mu)(\mu^{(k)} - \mu)^{\mathrm{T}}, \tag{38}$$

$$S_W = \sum_{k=1}^{c} \sum_{x_i \in X_k} (x_i - \mu^{(k)})(x_i - \mu^{(k)})^{\mathrm{T}}. \tag{39}$$

If we project the set on a one-dimensional space spanned by a given vector a, then the quantity

$$a^{\mathrm{T}} S_B a = \sum_{k=1}^{c} n_k |a^{\mathrm{T}}(\mu^{(k)} - \mu)|^2$$

represents a weighted sum of (squared) distances of the projection of the centroids of each set from the mean μ. At the same time, the quantity

$$a^{\mathrm{T}} S_W a = \sum_{k=1}^{c} \sum_{x_i \in X_k} |a^{\mathrm{T}}(x_i - \mu^{(k)})|^2$$

is the sum of the variances of each of the projected sets. An illustration is shown in Figure 7.

LDA projects the data so as to maximize the ratio of these two numbers:

$$\max_a \frac{a^{\mathrm{T}} S_B a}{a^{\mathrm{T}} S_W a}. \tag{40}$$

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

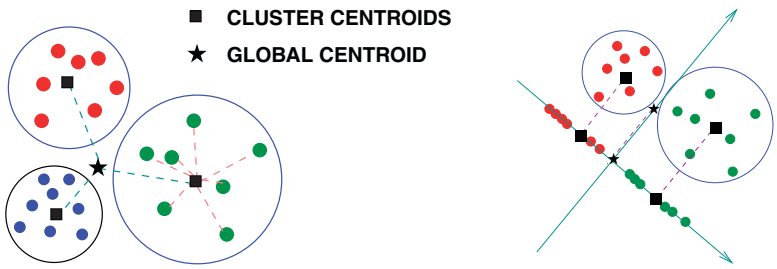

Figure 7. Left: class centroids and global centroid. Right: illustration of LDA in the one-dimensional case. Note that the main axis is shifted down for better clarity.

This optimal a is known to be an eigenvector associated with the largest eigenvalue of the pair (S_B, S_W). If we call S_T the total covariance matrix

$$S_T = \sum_{x_i \in X} (x_i - \mu)(x_i - \mu)^{\mathrm{T}} \tag{41}$$

then,

$$S_T = S_W + S_B. \tag{42}$$

Therefore, (40) is equivalent to

$$\max_a \frac{a^{\mathrm{T}} S_B a}{a^{\mathrm{T}} S_T a} \tag{43}$$

or

$$\min_a \frac{a^{\mathrm{T}} S_W a}{a^{\mathrm{T}} S_T a}, \tag{44}$$

where the optimal a is known to be an eigenvector associated with the largest eigenvalue of the pair (S_B, S_T), or the smallest eigenvalue of the pair (S_W, S_T).

The above one-dimensional projection generalizes to projections on d-dimensional spaces, i.e. we can modify the objective function such that the vector a is replaced by a matrix V. A traditional way toward such a generalization is to maximize the trace of $V^{\mathrm{T}} S_B V$ while requiring the columns of the solution matrix V to be S_W-orthogonal, i.e. imposing the condition $V^{\mathrm{T}} S_W V = I$. The optimum is achieved for the set of eigenvectors of the generalized eigenvalue problem

$$S_B u_i = \lambda_i S_W u_i$$

associated with the largest d eigenvalues. Incidentally, the above problem can also be formulated as a generalized singular value problem (see e.g. [33]). Another approach [22] casts the problem as maximizing the ratio of the two traces:

$$\max_{\substack{V \in \mathbb{R}^{n \times d} \\ V^{\mathrm{T}} V = I.}} \frac{\mathrm{Tr}[V^{\mathrm{T}} S_B V]}{\mathrm{Tr}[V^{\mathrm{T}} S_W V]},$$

Approaches for solving this problem were briefly discussed in Section 2.

Note that with simple algebraic manipulations, the matrices S_B, S_W and S_T can be expressed in terms of the data matrix $\bar{X}$:

$$S_B = \bar{X} H \bar{X}^{\mathrm{T}},$$

$$S_W = \bar{X}(I - H)\bar{X}^{\mathrm{T}},$$

$$S_T = \bar{X}\bar{X}^{\mathrm{T}}.$$

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

The matrix S_B has rank at most c because each of the blocks in H has rank one and therefore the matrix H itself has rank c. Because the matrix $I-H$ is an orthogonal projector, its range is the null-space of H which has dimension $n-c$. Thus, $I-H$, which plays the role of a Laplacean, has rank at most $n-c$. The corresponding eigenvalue problem to solve for (44) is

$$\bar{X}(I-H)\bar{X}^{\mathrm{T}}u_i=\lambda_i(\bar{X}\bar{X}^{\mathrm{T}})u_i. \tag{45}$$

We note finally that LDA can provide at most c meaningful projection directions, due to the fact that S_B has rank at most c. This may be too restrictive when one is interested in reduced spaces of dimensions larger than c.

6. SEMI-SUPERVISED DIMENSION REDUCTION

In many real-world applications of data mining, supervision information is hard to obtain. In other words, labeled data are typically few and labeling a huge set of data samples is tedious and time consuming. At the same time, there may be an abundance of unlabeled data which can be easily collected. In such a scenario, which is called semi-supervised, one is confronted with the challenge of exploiting both labeled and unlabeled data to solve the learning task.

Unsupervised methods may be insufficient for classification tasks, as shown on the left of Figure 8, where PCA provides misleading information about the best discriminant axis. On the other hand, supervised methods may face serious problems with over-fitting when the labeled data are very few. This is shown on the right of Figure 8, where LDA is biased by the particular instantiation of the small labeled set. Using the distribution of the unlabeled data, one could potentially remove the bias and 'correct' the LDA projection axis.

The main goal of semi-supervised methods is to use *both* labeled and unlabeled data in order to address such problems and achieve effective performance on future (out-of-sample) data points. In the past few years, semi-supervised methods for dimension reduction have attracted great interest. A short overview of the recent related research efforts is provided next.

6.1. Linear methods

A straightforward criterion for semi-supervised dimension reduction is the following

$$\max_{v}\frac{1}{2n^2}\sum_{i,j}(v^{\top}x_i-v^{\top}x_j)^2+\frac{\alpha}{2}\sum_{\ell(x_i)\neq\ell(x_j)}(v^{\top}x_i-v^{\top}x_j)^2-\frac{\beta}{2}\sum_{\ell(x_i)=\ell(x_j)}(v^{\top}x_i-v^{\top}x_j)^2,$$

where a one-dimensional projection has been considered for simplicity. In the above, $\ell(\cdot)$ denotes the class label of the data sample. The first summand expresses the variance of all (both labeled and unlabeled) data samples in the reduced space, which is exactly the same as in the PCA criterion. The second and third summands involve only the labeled data samples and represent the nearness

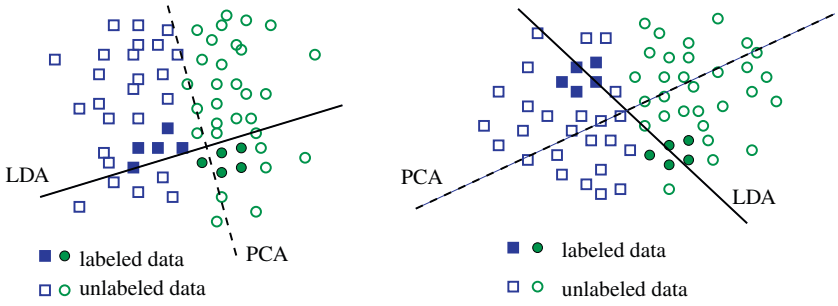

Figure 8. Illustrative examples of PCA and LDA weaknesses in semi-supervised settings. The solid line (resp. dashed line) denotes the LDA (resp. PCA) projection direction. The filled (resp. unfilled) symbols denote labeled (resp. unlabeled) data examples.

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

of samples from different and same classes, respectively. This criterion, which is closely related to that proposed in [34], makes use of both labeled and unlabeled data. Using similar derivations as in Section 4, the above criterion reads in the general d-dimensional case:

$$\max_{\substack{V\in\mathbb{R}^{m\times d}\\ V^{\mathrm{T}}V=I,}} \mathrm{Tr}[V^{\mathrm{T}}X(D-W)X^{\mathrm{T}}V], \tag{46}$$

where

$$W_{ij}=\begin{cases}\dfrac{1}{n^2}+\alpha, & \ell(x_i)\neq\ell(x_j)\\ \dfrac{1}{n^2}-\beta, & \ell(x_i)=\ell(x_j)\\ \dfrac{1}{n^2} & \text{otherwise}\end{cases}$$

and, as before, D is a diagonal matrix holding the row sums of W. Hence, the dimension reduction matrix V is obtained from the eigenvectors associated with the largest eigenvalues of an appropriately defined Laplacean matrix. Observe that in this case, the above semi-supervised criterion results in a modified weight matrix W with different weighting schemes for labeled and unlabeled data.

Several approaches have attempted to extend LDA to semi-supervised settings. Cai *et al.* in [35] proposed the so-called Semi-supervised Discriminant Analysis (SDA), which is a regularized variant of LDA that takes into account the unlabeled data. In particular, introducing a regularization term $J(a)$ into (43) results in

$$\max_{a}\frac{a^{\mathrm{T}}S_Ba}{a^{\mathrm{T}}S_Ta+\mu J(a)}. \tag{47}$$

The main idea in [35] is to include the manifold structure implied by all labeled and unlabeled data samples as an unsupervised regularizer in the above optimization problem. Hence, they proposed to use $J(a)=\sum_{ij}W_{ij}(a^{\top}x_i-a^{\top}x_j)^2$, where W is the 0/1 weight matrix of the affinity graph formed from all labeled and unlabeled data samples (i.e. $w_{ij}=1$ if i and j are adjacent, $w_{ij}=0$ otherwise). With this, (47) results in a generalized eigenvalue problem of the following form:

$$S_Ba=\lambda(S_T+\mu XLX^{\top})a,$$

where L is the corresponding Laplacean matrix of the affinity graph. Similar to LDA, this approach can only provide at most c projection directions. A very similar idea has been independently proposed in [36]. Another form of regularization for LDA has been proposed in [37], where the regularizer is defined from robust path-based similarities computed on the affinity graph. This approach also suffers from the limitation that the number of projection directions is bounded by the number of classes c.

A different approach for semi-supervised extension of LDA has been proposed in [38]. The main idea is to optimize the LDA criterion with respect to the labels of the unlabeled data. This is formulated as a concave minimization problem and the constrained concave-convex procedure (CCCP) is employed to solve it. After the solution of the optimization problem has been obtained, those unlabeled data samples whose class labels are estimated with high confidence are introduced in the original labeled set, and LDA is performed again in order to produce a more stable and more discriminant dimension reduction matrix.

Finally, the authors in [25] proposed a semi-supervised extension of orthogonal discriminant analysis using label propagation. The latter is used as a tool to obtain a soft label for each unlabeled data sample. Then the between-class and within-class scatter matrices are built according to those soft labels.

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

6.2. Nonlinear methods

The authors in [39] proposed the extension of the nonlinear dimension reduction methods LLE, ISOMAP and LTSA [12] to the semi-supervised setting. Unlike the methods previously described, selected points in the data set are assigned coordinates in the reduced dimension space (also known as 'on-manifold' coordinates) prior to the dimension reduction procedure. The study in [39] shows that the low-dimensional coordinates of the remainder of the samples can be obtained by solving a linear system of equations. In a recent work [40], the authors have revisited the same problem and they propose a spectral method to address it. They formulate a trace constraint capturing the fact that the estimates of the unknown low-dimensional vectors y_i should be close to the provided on-manifold coordinates. This is combined with the trace optimization problem of the LTSA method and the overall optimization results in a standard eigenvalue problem. However, it should be noted that such supervision information in the form of on-manifold coordinates, as is used in both approaches, is hard to obtain in practice.

7. CONNECTIONS BETWEEN DIMENSION REDUCTION METHODS

This section establishes connections between some of the methods discussed in the previous sections.

7.1. Relation between the LLE matrix and the Laplacean matrix

A comparison between (12) and (18) shows that the two are quite similar. The only difference is in the matrix inside the bracketed term. In one case it is of the form $Y(I-\hat{W})Y^{\mathrm{T}}$ where $I-\hat{W}$ is the normalized graph Laplacean, and in the other it is of the form $Y(I-W^{\mathrm{T}})(I-W)Y^{\mathrm{T}}$ where W is an affinity matrix. Can one just interpret the LLE matrix $(I-W^{\mathrm{T}})(I-W)$ as a Laplacean matrix? A Laplacean matrix L associated with a graph is a *a symmetric matrix whose off-diagonal entries are non-positive, and whose row-sums are zero* (or equivalently, the diagonal entries are the negative sums of the off-diagonal entries). In other words, $l_{ij}\leqslant 0$ for $i\neq 0$, $l_{ii}=-\sum_j l_{ij}$. The LLE matrix $M=(I-W)^{\mathrm{T}}(I-W)$ satisfies the second property (zero row sum) but not the first (non-positive off-diagonals) in general.

Proposition 7.1
The symmetric matrix $M=(I-W)^{\mathrm{T}}(I-W)$ has zero row (and column) sums. In addition, denoting by $w_{:j}$ the j-th column of W,

$$m_{jj}=1+\|w_{:j}\|^2,\quad m_{ij}=-(w_{ij}+w_{ji})+\langle w_{:j},w_{:i}\rangle,\quad i\neq j. \tag{48}$$

Proof
As $(I-W)$ has row sums equal to zero, then $(I-W)\mathbf{1}=0$ and therefore $M\mathbf{1}=(I-W^{\mathrm{T}})(I-W)\mathbf{1}=0$, which shows that the row sums of M are zero. As M is symmetric, its column-sums are also zero. Since $M=I-W-W^{\mathrm{T}}+W^{\mathrm{T}}W$, a generic entry $m_{ij}=e_i^{\mathrm{T}}Me_j$ of M is given by,

$$\begin{aligned} m_{ij} &= e_i^{\mathrm{T}}e_j-e_i^{\mathrm{T}}We_j-e_i^{\mathrm{T}}W^{\mathrm{T}}e_j+e_i^{\mathrm{T}}W^{\mathrm{T}}We_j \\ &= \delta_{ij}-(w_{ij}+w_{ji})+\langle w_{:i},w_{:j}\rangle \end{aligned}$$

from which relations (48) follow immediately after recalling that $w_{ii}=0$. □

Expression (48) shows that the off-diagonal entries of M can be positive, i.e. it is not true that $m_{ij}\leqslant 0$ for all $i\neq j$. In the particular situation when $w_{ij}=w_{ji}=0$ and $i\neq j$, then $m_{ij}=\langle w_{:i},w_{:j}\rangle$ and (48) implies that $m_{ij}\geqslant 0$. When w_{ij} and w_{ji} are not both equal to zero but they are both small, then by the same argument it is likely that m_{ij} will be non-negative. It can be observed with randomly generated sparse matrices that in general there are few other instances of positive off-diagonal

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

entries, i.e. in most cases, m_{ij} is positive only when $w_{ij}+w_{ji}$ is zero or small. For example, for the matrix

$$W = \begin{pmatrix} 0 & 0.4 & 0.6 & 0 \\ 0.1 & 0 & 0.3 & 0.6 \\ 0.2 & 0.4 & 0 & 0.4 \\ 0 & 0.5 & 0.5 & 0 \end{pmatrix}$$

one finds that all off-diagonal entries of $(I-W^{\mathrm{T}})(I-W)$ are negative except the entries (1,4) and (by symmetry) (4,1) whose value, the inner product of columns 1 and 4, equals 0.14.

Among other similarities between the LLE matrix and the graph Laplacean is the fact that both matrices are symmetric positive semi-definite and that they are both related to the local structure of the data since they relate the nearby samples by a relation.

As not every matrix $M=(I-W^{\mathrm{T}})(I-W)$ can be a graph Laplacean matrix, one can ask the reverse question: Given a normalized Laplacean matrix which we write as $\hat{L}=I-\hat{W}$, is it possible to find a matrix W such that the matrix M equals $\hat{L}$? One easy answer is obtained by restricting W to being symmetric. In this case, $W=I-\sqrt{I-\hat{W}}$, which is dense and not necessarily positive. There is one important situation where the Graph Laplacean is easily written as an LLE matrix and that is when $I-W$ is a projector. One specific situation of interest is when $L=I-\frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}$, which is the projector used by PCA, see (22). In this case $(I-W^{\mathrm{T}})(I-W)=I-\hat{W}$ which means that the two methods will yield the same result. Yet another situation of the same type in which L is a projector, arises in supervised learning, which brings us to the next connection.

7.2. Connection between LDA, supervised NPP, and supervised LPP

Note that in the supervised setting discussed in Section 5.1, the block diagonal adjacency matrix H (see Equation (37)) is a projector. To see why this is true, define the characteristic vector g_k for class k as the vector of $\mathbb{R}^n$ whose ith entry is one if x_i belongs to class k and zero otherwise. Then H can be alternatively written as

$$H=\sum_{k=1}^{c}\frac{g_k g_k^{\mathrm{T}}}{n_k},$$

which shows that H is a projector. Now take $W=\hat{W}=H$ and observe that $(I-W^{\mathrm{T}})(I-W)=I-W=I-\hat{W}=I-H$ in this case. Next, compare (45), (31), and (35) and note that they are identical.

Proposition 7.2
LDA, supervised LPP, and supervised NPP are mathematically equivalent when $W=\hat{W}=H$.

7.3. Connection between PCA and LPP

Next we will make other important connections between PCA and LPP. One of these connections was observed in [28], see also [16]. Essentially, by defining the Laplacean graph to be a dense graph, specifically by defining $L=I-\frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}$, one can easily see that the matrix XLX^{T} is a scaled covariance matrix and thus *ignoring the constraint in LPP*, one would get the projection on the lowest modes instead of the highest ones as in PCA.

Another connection is now considered. Compare the two eigenproblems (24) and (31) and notice that for PCA we seek the largest eigenvalues, whereas for LPP we seek the smallest ones. If we are able to select $\hat{W}$ in (31) so that $\hat{X}(I-\hat{W})\hat{X}^{\mathrm{T}}=I$ then we would recover the result of PCA (apart from the diagonal scaling with D). We can restrict the choice by assuming $D=I$ and assume that the data are centered, hence $X\mathbf{1}=0$. Then it is easy to select such a matrix $\hat{W}$ in the common situation where $m<n$ and X is of full rank. It is the matrix $\hat{W}=I-X^{\mathrm{T}}(XX^{\mathrm{T}})^{-2}X$. With

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

this, the LPP problem (31) becomes $v_i = \lambda_i(XX^{\mathrm{T}})v_i$ and we are computing the smallest λ_i and associated v_i's, which correspond to the largest eigenpairs of the covariance matrix. Note also that $I - \hat{W} = SS^{\mathrm{T}}$, where $S = X^{\dagger}$ is the pseudo-inverse of X. We will revisit this viewpoint when we discuss kernels in Section 8.

Proposition 7.3
When X is $m \times n$ with $m < n$ and full rank, LPP with the graph Laplacean replaced by the matrix $I - \hat{W} = X^{\mathrm{T}}(XX^{\mathrm{T}})^{-2}X$ is mathematically equivalent to PCA.

7.4. Connection to projection methods for eigenvalue problems

Comparing the eigenvalue problems (31) and (35) will reveal an interesting connection with projection methods for eigenvalue problems. Readers familiar with projection methods will recognize in these problems, a projection-type technique for eigenvalue problems, using the space spanned by X^{T}. Recall that a projection method for computing approximate eigenpairs of a matrix eigenvalue problem of the form

$$Au = \lambda u$$

utilizes a certain subspace $\mathcal{K}$ from which the eigenvectors are extracted. Specifically, the conditions are as follows, where the tildes denote the approximation: Find $\tilde{u} \in \mathcal{K}$ and $\tilde{\lambda} \in \mathbb{C}$ such that

$$A\tilde{u} - \tilde{\lambda}\tilde{u} \perp \mathcal{K}. \tag{49}$$

This is referred to as an orthogonal projection method. Stating that $\tilde{u} \in \mathcal{K}$ gives k degrees of freedom if $dim(\mathcal{K}) = k$, and condition (49) imposes k independent constraints. If V is a basis of the subspace $\mathcal{K}$, then the above conditions become $\tilde{u} = Vy$, for a certain $y \in \mathbb{R}^k$, and (49) leads to

$$V^{\mathrm{T}}(A - \tilde{\lambda}I)Vy = 0 \quad \text{or} \quad V^{\mathrm{T}}AVy = \tilde{\lambda}V^{\mathrm{T}}Vy.$$

LLE is mathematically equivalent to computing the lowest eigenspace of the LLE matrix $M = (I - W^{\mathrm{T}})(I - W)$. Eigenmaps seek the lowest eigenspace of the matrix $I - \hat{W}$.

Proposition 7.4
LPP is mathematically equivalent to a projection method on Span$\{X^{\mathrm{T}}\}$ applied to the normalized Laplacean matrix $\hat{L} = I - \hat{W}$, i.e. it is a projected version of eigenmaps. It will yield the exact result as eigenmaps when Span$\{X^{\mathrm{T}}\}$ is invariant under $\hat{L}$. NPP is mathematically equivalent to a projection method on Span$\{X^{\mathrm{T}}\}$ applied to the matrix $(I - W^{\mathrm{T}})(I - W)$, i.e. it is a projected version of LLE. It will yield the exact results as LLE when Span$\{X^{\mathrm{T}}\}$ is invariant under $(I - W^{\mathrm{T}})(I - W)$.

One particular case when the two methods will be mathematically equivalent is in the special situation of undersampling, i.e. when $m \geqslant n$ and the rank of X is equal to n. In this case X^{T} is of rank n and therefore the subspace Span$\{X^{\mathrm{T}}\}$ is trivially invariant under $\hat{L}$.

Corollary 7.5
When the column rank of X is equal to n (undersampled case) LPP is mathematically equivalent to Eigenmaps and NPP is mathematically equivalent to LLE.

7.5. Connection to spectral clustering/partitioning

It is important to comment on a few relationships with the methods used for spectral clustering (graph partitioning) [41–44]. Given a weighted undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a k-way partitioning amounts to finding k disjoint subsets $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_k$ of the vertex set $\mathcal{V}$ so that the total weights of the edges that cross different partitions are minimized while the sizes of the subsets are roughly balanced. Formally, a k-way clustering minimizes the cost function:

$$\mathcal{F}(\mathcal{V}_1, \ldots, \mathcal{V}_k) = \sum_{\ell=1}^{k} \frac{\sum_{i \in \mathcal{V}_\ell, j \in \mathcal{V}_\ell^c} w_{ij}}{\sum_{i \in \mathcal{V}_\ell} d_i}, \tag{50}$$

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

where $d_i = \sum_{j \in \mathcal{V}} w_{ij}$ is the degree of a vertex i. For each term in the summation of this objective function, the numerator $\sum_{i \in \mathcal{V}_\ell, j \in \mathcal{V}_\ell^c} w_{ij}$ is the sum of the weights of edges crossing the partition $\mathcal{V}_\ell$ and its complement $\mathcal{V}_\ell^c$, while the denominator $\sum_{i \in \mathcal{V}_\ell} d_i$ is the 'size' of the partition $\mathcal{V}_\ell$.

If we define an $n \times k$ matrix Z whose ℓ-th column is a cluster indicator of the partition $\mathcal{V}_\ell$, i.e.

$$Z(j, \ell) = \begin{cases} 1 \Big/ \sqrt{\sum_{i \in \mathcal{V}_\ell} d_i} & \text{if } j \in \mathcal{V}_\ell \\ 0 & \text{otherwise,} \end{cases} \tag{51}$$

then the cost function is exactly the trace of the matrix $Z^{\mathrm{T}} L Z$

$$\mathcal{F}(\mathcal{V}_1, \ldots, \mathcal{V}_k) = \mathrm{Tr}(Z^{\mathrm{T}} L Z)$$

with Z satisfying

$$Z^T D Z = I,$$

where L (the graph Laplacean) and D are defined as before. Therefore, the clustering problem stated above can be formulated as the problem of finding a matrix Z in the form of (51) such that $\mathrm{Tr}(Z^T L Z)$ is minimum and $Z^T D Z = I$. This being a hard problem to solve, one usually considers a heuristic which computes a matrix Z that is no longer restricted to the form (51), so that the same two conditions are still satisfied. With this relaxation, the columns of Z are known to be the k smallest eigenvectors of the generalized eigenvalue problem

$$L z_i = \lambda_i D z_i. \tag{52}$$

The above solution Z has a natural interpretation related to Laplacean Eigenmaps. Imagine that there is a set of high-dimensional data samples lying on a manifold. We perform dimension reduction on these data samples using the Laplacean Eigenmaps method. Then Z is the low-dimensional embedding of the original manifold, that is, each sample on the manifold is mapped to a row of Z, in the k-dimensional space. Thus, a good clustering of Z in some sense implies a reasonable clustering of the original high-dimensional data.

It is worthwhile to mention that by slightly modifying the cost function (50) we can arrive at a similar spectral problem. For this, consider minimizing the objective function

$$\hat{\mathcal{F}}(\mathcal{V}_1, \ldots, \mathcal{V}_k) = \sum_{\ell=1}^{k} \frac{\sum_{i \in \mathcal{V}_\ell, j \in \mathcal{V}_\ell^c} w_{ij}}{|\mathcal{V}_\ell|}. \tag{53}$$

Comparing (53) with (50), one sees that the only difference in the objective is the notion of 'size of a subset': here the number of vertices $|\mathcal{V}_\ell|$ is used to measure the size of $\mathcal{V}_\ell$, while in (50) this is replaced by the sum of the degree of the vertices in $\mathcal{V}_\ell$, which is related to the number of edges. Similar to the original problem, if we define the matrix $\hat{Z}$ as

$$\hat{Z}(j, \ell) = \begin{cases} 1/\sqrt{|\mathcal{V}_\ell|} & \text{if } j \in \mathcal{V}_\ell \\ 0 & \text{otherwise} \end{cases}$$

then we get the following two equations:

$$\hat{\mathcal{F}}(\mathcal{V}_1, \ldots, \mathcal{V}_k) = \mathrm{Tr}(\hat{Z}^T L \hat{Z}), \quad \hat{Z}^{\mathrm{T}} \hat{Z} = I.$$

The cost function (53) is again hard to minimize and we can relax the minimization to obtain the eigenvalue problem:

$$L \hat{z}_i = \hat{\lambda}_i \hat{z}_i. \tag{54}$$

The partitioning resulting from minimizing the objective function (53) approximately via (54) is called the *ratio cut* [45]. The one resulting from minimizing (50) approximately via (52) is called the *normalized cut* [41]. We will refer to the problem of finding the ratio cut (resp. finding the normalized cut), as the *spectral ratio cut problem*, (resp. *spectral normalized cut problem*).

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

Table I. Objective functions and constraints used in several dimension reduction methods.

Method	Object. (min)	Constraint
LLE	$\mathrm{Tr}[Y(I-W^{\mathrm{T}})(I-W)Y^{\mathrm{T}}]$	$YY^{\mathrm{T}}=I$
Eigenmaps	$\mathrm{Tr}[Y(D-W)Y^{\mathrm{T}}]$	$YDY^{\mathrm{T}}=I$
PCA/MDS	$\mathrm{Tr}[-V^{\mathrm{T}}X(I-\frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}})X^{\mathrm{T}}V]$	$V^{\mathrm{T}}V=I$
LPP	$\mathrm{Tr}[V^{\mathrm{T}}X(D-W)X^{\mathrm{T}}V]$	$V^{\mathrm{T}}XDX^{\mathrm{T}}V=I$
OLPP	$\mathrm{Tr}[V^{\mathrm{T}}X(D-W)X^{\mathrm{T}}V]$	$V^{\mathrm{T}}V=I$
NPP	$\mathrm{Tr}[V^{\mathrm{T}}X(I-W^{\mathrm{T}})(I-W)X^{\mathrm{T}}V]$	$V^{\mathrm{T}}XX^{\mathrm{T}}V=I$
ONPP	$\mathrm{Tr}[V^{\mathrm{T}}X(I-W^{\mathrm{T}})(I-W)X^{\mathrm{T}}V]$	$V^{\mathrm{T}}V=I$
LDA	$\mathrm{Tr}[V^{\mathrm{T}}X(I-H)X^{\mathrm{T}}V]$	$V^{\mathrm{T}}XX^{\mathrm{T}}V=I$
Spect. clust. (ratio cut)	$\mathrm{Tr}[Z^{\mathrm{T}}(D-W)Z]$	$Z^{\mathrm{T}}Z=I$
Spect. clust. (normalized cut)	$\mathrm{Tr}[Z^{\mathrm{T}}(D-W)Z]$	$Z^{\mathrm{T}}DZ=I$

Finding the ratio cut amounts to solving the standard eigenvalue problem related to the graph Laplacean L, while finding the normalized cut is equivalent to solving the eigenvalue problem related to the normalized Laplacean $\hat{L}=D^{-1/2}LD^{-1/2}$. This connection results from different interpretations of the 'size of a set'. The second smallest eigenvector $\hat{z}_2$ (the *Fiedler vector* [46, 47]) of L plays a role similar to that of vector z_2 described above. As Z is the standard low-dimensional embedding of the manifold in the high-dimensional ambient space, a natural question is: Is $\hat{Z}$ also a good embedding of this manifold? As will be seen in Section 8.2, $\hat{Z}$ is the low-dimensional embedding of a 'kernel' version of PCA that uses an appropriate kernel.

7.6. Unifying framework

We now summarize the various connections that we have drawn so far. The objective functions and the constraints imposed on the optimization problems seen so far are shown in Table I. As can be seen, the methods can be split into two classes. The first class, which can be termed a class of 'implicit mappings', includes LLE, Laplacean Eigenmaps and ISOMAP. Here, one obtains the low-dimensional data set Y by solving an optimization problem of the form

$$\min_{\substack{Y\in\mathbb{R}^{d\times n}\\ YBY^{\mathrm{T}}=I.}} \mathrm{Tr}[YAY^{\mathrm{T}}], \tag{55}$$

where B is either the identity matrix (LLE) or the matrix D (Eigenmaps). For LLE the matrix A is $A=(I-W^{\mathrm{T}})(I-W)$ and for Eigenmaps, A is the Laplacean matrix.

The second class of methods, which can be termed the class of 'projective mappings' includes PCA/MDS, LPP, ONPP, and LDA, and it can be cast as an optimization problem of the form

$$\min_{\substack{V\in\mathbb{R}^{m\times d}\\ V^{\mathrm{T}}BV=I.}} \mathrm{Tr}[V^{\mathrm{T}}XAX^{\mathrm{T}}V], \tag{56}$$

Here, B is either the identity matrix (ONPP, PCA) or a matrix of the form XDX^{T} or XX^{T}. For ONPP, the matrix A is the same as the LLE matrix $(I-W)(I-W^{\mathrm{T}})$, and for LPP, A is a Laplacean graph matrix. For LDA, $A=I-H$. For PCA/MDS the largest eigenvalues are considered hence the trace is maximized instead of minimized. This means that we need to take A to be the negative identity matrix for this case. In all cases the resulting V matrix is the projector, hence $Y=V^{\mathrm{T}}X$ is the low-dimension data. Figure 9 shows pictorially the relations between the various dimension reduction methods.

8. KERNELS

Kernels have been extensively used as a means to represent data by mappings that are intrinsically nonlinear, see, e.g. [48–51]. Kernels are based on an implicit nonlinear mapping $\Phi:\mathbb{R}^m\rightarrow\mathscr{H}$,

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

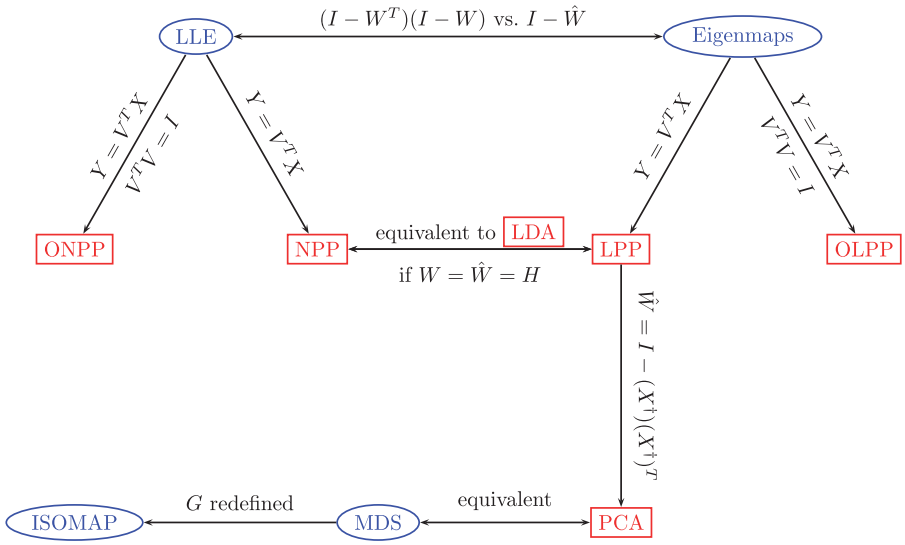

Figure 9. Relationships between the different dimension reduction methods.

where $\mathscr{H}$ is a certain high-dimensional *feature space*. Denote by $\Phi(X)=[\Phi(x_1),\Phi(x_2),\ldots,\Phi(x_n)]$ the transformed data set in $\mathscr{H}$. We will also use Φ (a matrix) as a shorthand notation for $\Phi(X)$ when there is no risk of confusion with the mapping.

The Moore–Aronszajn theorem [52] asserts that every symmetric positive-definite kernel is associated with a dot product defined on some Hilbert space. As a result, for finite samples X, the transformation Φ need only be known through its Grammian, which is symmetric positive (semi-)definite, on the data X. In other words, what is known is the matrix K whose entries are

$$K_{ij}\equiv k(x_i,x_j)=\langle\Phi(x_i),\Phi(x_j)\rangle. \tag{57}$$

This is the Gram matrix induced by the kernel $k(x,y)$ associated with the feature space. In fact, another interpretation of the kernel mapping is that we are defining an alternative inner product in the X-space, which is expressed through the inner product of every pair (x_i,x_j) as $\langle x_i,x_j\rangle=k_{ij}$.

Formally, any of the techniques seen so far can be implemented with kernels as long as its inner workings require only inner products to be implemented. In the sequel we denote by K the kernel matrix:

$$K\equiv\Phi(X)^{\mathrm{T}}\Phi(X)=[k_{i,j}]_{i,j=1,\ldots,n}=[\Phi(x_i)^{\mathrm{T}}\Phi(x_j)]_{i,j=1,\ldots,n}. \tag{58}$$

8.1. Explicit mappings with kernels

Consider now the use of kernels in the context of the 'projective mappings' seen in Section 7.6. These compute a projection matrix V by solving an optimization problem of the form (56). Formally, if we were to work in feature space, then X in (56) would become Φ, i.e. the projected data would take the form $Y=V^{\mathrm{T}}\Phi$. Here $V\in\mathbb{R}^{N\times d}$, where N is the (typically large and unknown) dimension of the feature space.

The cost function (56) would become

$$\mathrm{Tr}[V^{\mathrm{T}}\Phi A\Phi^{\mathrm{T}}V], \tag{59}$$

where A is one of the matrices defined earlier for each method. We note in passing that the matrix A, which should capture local neighborhoods, must be based on the data and the distances between them in the feature space.

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

As Φ is not explicitly known (and is of large dimension) this direct approach does not work. However, as was suggested in [29], one can exploit the fact that V can be restricted (again implicitly) to lie in the span of Φ, since V must project Φ. For example, we can implicitly use an orthogonal basis of the span of Φ via an implicit QR factorization of Φ, as was done in [29]. In the following, this factorization is avoided for simplicity.

8.2. Kernel PCA

Kernel PCA, see, e.g. [53], corresponds to performing classical PCA on the set $\{\Phi(x_i)\}$. Using $\bar{\Phi}$ to denote the matrix $[\Phi(\bar{x}_1), \ldots, \Phi(\bar{x}_n)]$, this leads to the optimization problem:

$$\max \mathrm{Tr}[V^{\mathrm{T}}\bar{\Phi}\bar{\Phi}^{\mathrm{T}}V] \quad \text{s.t. } V^{\mathrm{T}}V = I.$$

From what was seen before, we would need to solve the eigenvalue problem

$$\bar{\Phi}\bar{\Phi}^{\mathrm{T}}u_i = \lambda u_i$$

and the projected data will be $Y = [u_1, \ldots, u_d]^{\mathrm{T}}\bar{\Phi}$.

The above problem is not solvable as is because the matrix $\bar{\Phi}\bar{\Phi}^{\mathrm{T}}$ is not readily available. What is available is the Grammian $\bar{\Phi}^{\mathrm{T}}\bar{\Phi}$. This suggests the following right singular vector approach. We multiply both sides of the above equation by $\bar{\Phi}^{\mathrm{T}}$, which yields:

$$\underbrace{[\bar{\Phi}^{\mathrm{T}}\bar{\Phi}]}_{\bar{K}}\bar{\Phi}^{\mathrm{T}}u_i = \lambda_i\bar{\Phi}^{\mathrm{T}}u_i.$$

We stated above that the matrix $\bar{K}$ is available – but in reality since the Φ_i are not explicitly available we cannot recenter the data in feature space. However, there is no real issue because $\bar{K}$ can be expressed easily from K since $\bar{K} = \bar{\Phi}^{\mathrm{T}}\bar{\Phi} = (I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}})K(I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}})$, see [48].

Recall that $Y = V^{\mathrm{T}}\bar{\Phi}$, where $V = [u_1, \ldots, u_d]$, hence the vectors $\bar{\Phi}^{\mathrm{T}}u_i$ in the above equation are just the transposes of the rows of the low-dimensional Y. In the end, the rows of Y, when transposed, are the largest d eigenvectors of the Gram matrix. In other words, Y is obtained by solving the largest d eigenvectors of the system

$$\bar{K}z_i = \lambda_i z_i, \quad [z_1, \ldots, z_d] = Y^{\mathrm{T}}. \tag{60}$$

It is interesting to compare this problem with the one obtained for the spectral ratio cut (54): the columns of Y^{T} (n-vectors) are the smallest eigenvectors of the Laplacean matrix L. Hence, it is clear that the spectral ratio cut problem can be interpreted as Kernel PCA with the kernel matrix $K = L^{\dagger}$ [17, 54].

Proposition 8.1
The kernel version of PCA, using the kernel matrix $K = L^{\dagger}$, is mathematically equivalent to the spectral ratio cut problem in feature space.

8.3. Kernel LPP

To define a kernel version of LPP, we can proceed similar to PCA. Denote again by Φ the system $\Phi \equiv \Phi(X)$, and let $K \equiv \Phi^{\mathrm{T}}\Phi$, which is assumed to be invertible. The problem (28) for LPP in feature space is

$$\min_V \mathrm{Tr}[V^{\mathrm{T}}\Phi L\Phi^{\mathrm{T}}V] \quad \text{s.t. } V^{\mathrm{T}}\Phi D\Phi^{\mathrm{T}}V = I,$$

which leads to the eigenvalue problem:

$$\Phi L\Phi^{\mathrm{T}}u_i = \lambda_i\Phi D\Phi^{\mathrm{T}}u_i.$$

Again this is not solvable because the matrices $\Phi L\Phi^{\mathrm{T}}$ and $\Phi D\Phi^{\mathrm{T}}$ are not available.

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

Proceeding in the same way as for PCA, and assuming for simplicity that Φ is of full rank, we can left-multiply by Φ^{T}, then by K^{-1}, and recalling that $Y = V^{\mathrm{T}}\Phi$, we obtain $Y^{\mathrm{T}} = [z_2, \ldots, z_{d+1}]$ where

$$Lz_i = \lambda_i Dz_i. \tag{61}$$

One may be puzzled by the remarkable fact that the Grammian matrix K no longer appears in the equation. It is important to recall however, that the information about distances must already be reflected in the Laplacean pair (L, D). This fact is discussed in more detail in [28].

Proposition 8.2
The kernel version of LPP is mathematically equivalent to Laplacean eigenmaps in feature space.

We note that this is in fact a practical equivalence as well, i.e. the computational problems at which the two methods arrive are the same. What appeared to be a nonlinear method (eigenmaps) becomes a linear one using a kernel.

An immediate question is: do we explicitly know the related mapping? In [55], an infinite dimensional operator was used as a means to define out-of-sample extensions of various nonlinear methods. All that is needed is to find a continuous kernel $k(x, y)$ whose discretization gives rise to the discrete kernel $k(x_i, x_j)$.

8.4. Kernel ONPP

The kernel version of ONPP seeks to minimize the function

$$\min_{V \in \mathbb{R}^{N\times d}, V^{\mathrm{T}}V = I} \quad [V^{\mathrm{T}}\Phi M \Phi^{\mathrm{T}} V], \tag{62}$$

which leads to the eigenvalue problem:

$$\Phi M \Phi^{\mathrm{T}} u_i = \lambda_i u_i. \tag{63}$$

We now again multiply by Φ^{T} to the left and note as before $K = \Phi^{\mathrm{T}}\Phi$, and that the solution Y is such that $Y^{\mathrm{T}} = \Phi^{\mathrm{T}}[u_2, \ldots, u_{d+1}]$. This leads to the eigenvalue problem

$$KMz_i = \lambda_i z_i \quad \text{or} \quad Mz_i^{\mathrm{T}} = K^{-1} z_i^{\mathrm{T}}, \quad [z_2, \ldots, z_{d+1}] = Y^{\mathrm{T}} \tag{64}$$

whose solution is the set of eigenvectors of the matrix M but with a different orthogonality constraint, namely the K^{-1}-orthogonality. In other words, the rows of the projected data Y can be directly computed as the (transposed) eigenvectors of the matrix KM associated with the smallest d eigenvalues.

Although the matrix KM in (64) is non-symmetric, the problem is similar to the eigenvalue problem $Mz = \lambda K^{-1} z$ and therefore, the eigenvectors are orthogonal with respect to the K^{-1}-inner product, i.e. $z_i^{\mathrm{T}} K^{-1} z_j = \delta_{ij}$. This can also be seen by introducing the Cholesky factorization of K, $K = RR^{\mathrm{T}}$ and setting $\hat{z} = R^{-1}z$. The set of $\hat{z}$'s is orthogonal.

It is also useful to translate the optimization problem corresponding to the eigenvalue problem (64) for the Y variable. Clearly Kernel ONPP solves the optimization problem:

$$\min_{\substack{Y \in \mathbb{R}^{d\times n} \\ YK^{-1}Y^{\mathrm{T}} = I.}} \mathrm{Tr}[YMY^{\mathrm{T}}], \tag{65}$$

This new problem is again in $\mathbb{R}^n$. In practice, there is still an issue to be resolved with this new setting, namely we need a matrix $M = (I - W^{\mathrm{T}})(I - W)$ which is determined for the samples in feature space. In other words the affinity matrix W should be for the samples $\Phi(x_i)$ not the x_i's. Again this is easily achievable because the method for constructing W only requires local Grammians, which are available from K; see [16] for details.

We now address the same question as the one asked for the relation between LPP and eigenmaps in feature space. The question is whether or not performing LLE in feature space will yield the

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

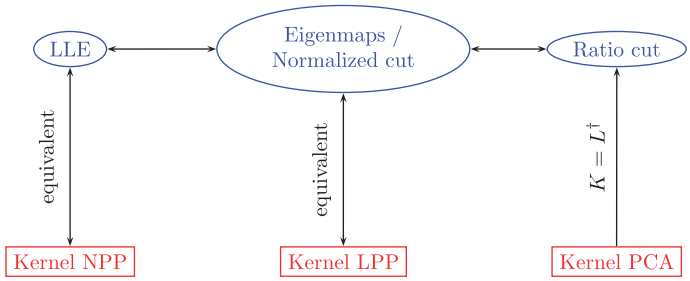

Figure 10. Kernel methods and their equivalents.

kernel version of ONPP. Clearly, the problem (65) to which we arrive with kernel ONPP does not resemble the optimization problem of LLE. This is easy to understand: ONPP uses an orthogonal projection whereas LLE requires the embedded data to be orthogonal. If we were to enforce the same orthogonality on the y_i's as in LLE we might obtain the same result, and this is indeed the case.

Recall that we defined this option in Section 4.5 and called it NPP. Consider this alternative way of defining ONPP and referred to as NPP in Section 4.5. Proceeding as above, one arrives at the following optimization problem for Kernel NPP:

$$\min_{V \in \mathbb{R}^{N \times d}, V^{\mathrm{T}} \Phi \Phi^{\mathrm{T}} V = I} [V^{\mathrm{T}} \Phi M \Phi^{\mathrm{T}} V],$$

which leads to the eigenvalue problem:

$$\Phi M \Phi^{\mathrm{T}} u_i = \lambda_i \Phi \Phi^{\mathrm{T}} u_i .$$

Multiplying by Φ^{T} and then by K^{-1}, we arrive again at the following problem from which the kernel matrix K has again disappeared:

$$M \Phi^{\mathrm{T}} u_i = \lambda_i \Phi^{\mathrm{T}} u_i \rightarrow M z_i = \lambda_i z_i . \tag{66}$$

The projected data is now identical to that obtained from LLE applied to Φ.

Proposition 8.3
Kernel NPP is equivalent to LLE performed in feature space.

Figure 10 summarizes pictorially the relations that have been revealed in this section.

It is interesting to note that kernel methods tend to use dense kernels—as these are commonly defined as integral operators. Graph Laplaceans on the other hand are sparse and represent inverses of integral operators. This is exactly the same situation one has with operators on Hilbert spaces: kernel operators are compact operators which when discretized yield dense matrices (e.g. by the Nystrom method), and their inverses are partial differential operators which when discretized yield sparse matrices.

8.5. What about LLE and eigenmaps?

In principle, it would be perfectly possible to implement kernel variants of LLE and eigenmaps—since these require constructions of neighborhood matrices which can be adapted by using distances obtained from some Grammian K. However, this would be redundant with the nonlinear nature of LLE/eigenmaps. To understand this it is useful to come back to the issue of the similarity of LLE with Kernel ONPP. Comparing the two methods, one observes that the eigenvalue problems of the projective methods (PCA, LPP, ONPP, ...) are $m \times m$ problems, i.e. they are in the data space. In contrast, all kernel methods share with LLE and eigenmaps the fact that the eigenproblems are all $n \times n$. Thus, none of the eigenvalue problems solved by Kernel PCA, Kernel LPP, and Kernel ONPP involves the data set X explicitly, in contrast with those eigenvalue problems seen

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

for the non-kernel versions of the same methods. Compare for example (29) for the standard LPP with (61) for Kernel LPP or the problems (24) and (60) for PCA and kernel PCA. In essence, the data are hidden in the Gram matrix K (or its Cholesky factor R) for PCA, and/or the Laplacean pair L, D for LPP. In effect, one can consider that there is only one big class of methods which can be defined using various kernels.

We conclude that the linear and nonlinear families of methods can be brought together thanks to the use of kernels. The observation that kernels can help unify dimension reduction has been made before. Ham *et al.* [17] note that several of the known standard methods (LLE [9], Isomap [13], Laplacean eigenmaps [11, 27]) can be regarded as some form of Kernel PCA. In [16], it was observed that linear and nonlinear projection methods are in fact equivalent, in the sense that one can define one from the other with the help of kernels.

8.6. The kernel effect: A toy example

To illustrate the power of kernels, it is best to take a small artificial example. We randomly draw 250 points from a square of width 1.5 centered at the origin, and 250 additional points from an annulus surrounding the square. In particular, the annulus is defined as the region between a half disk of radius 3.5 and a half disk of radius 4.5, both centered at $[1, 0]$. This is shown in the first plot of Figure 11. The figure is in 2-D. The line shown in this first figure shows how a method based on PCA (called PDDP, see [56]) partitions the set. It fails to see the two distinct parts. In fact any linear separation will do a mediocre job here because the two sets cannot be partitioned by a straight line. What we do next is to use kernels to transform the set. In fact the experiment is unusual whereas we take the 2-D set and project it into a 2-D set with Kernel PCA. Recall that this is equivalent to eigenmaps with the Grammian matrix replacing the usual graph Laplacean. The method amounts to simply taking the kernel $\bar{K}$ (see Section 8.2 and Equation (60)) and computing its largest 2 eigenvectors. This yields two vectors which after transposition yield the projected data Y. As the dimensions of X and Y are the same there is no dimension reduction *per se*, but the projection will nevertheless show the effect of kernels and illustrate how they work.

We use a Gaussian (or heat) kernel which we write in the form $K(x, y) = \exp(-\|x - y\|_2^2/\sigma^2)$. This is a very popular kernel, see, e.g. [57]. One of the difficulties with this kernel is that it requires finding a good parameter σ. It is often suggested to select a value of σ equal to half the median of pairwise distances obtained from a large sample of points. In our case, we use all the 500 points for this purpose and call σ_0 the corresponding optimal value. In the experiment we use several values of σ around this pseudo-optimal value σ_0. Specifically we take σ^2 of the form σ_0^2/C, where C takes the values: $C = 3, 2, 1, 0.5, 0.2$. The results of the related KPCA projections are shown in Figure 11.

When the parameter C takes values of 0.1 ($\sigma^2 \approx 27.46..$) and smaller, the resulting figures begin to resemble the original picture. These are omitted. This experiment reveals a number of features of kernel methods in general and this kernel in particular. When σ is large (C in the experiment is small), then the inner-products become basically close to being constant (constant one) and hence the Grammian will then be similar to the trivial one seen for PCA. This means we will tend to get results similar to those with standard PCA, and this is indeed what is observed. For smaller values of σ the situation is quite different. In this case, large pairwise squared distances $\|x - y\|^2$ are amplified and the negative exponential essentially makes them close to zero. This has the effect of 'localizing' the data. For $\sigma = \sigma_0$, (leftmost figure in second row), the separation achieved between the two sets is quite remarkable. Now an algorithm such as K-means (see, e.g. [1]) can do a perfect job of identifying the two clusters (provided we know there are two such clusters) and a linear separation can also be easily achieved. This is a major reason that linear methods are not to be neglected. Note that as σ increases, the set corresponding to the annulus expands gradually from a very densely clustered set to one which reaches a better balance with the other set (for σ_0 for example). This can be explained by the fact that pairwise distances between points of the annulus are larger than those of the square.

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

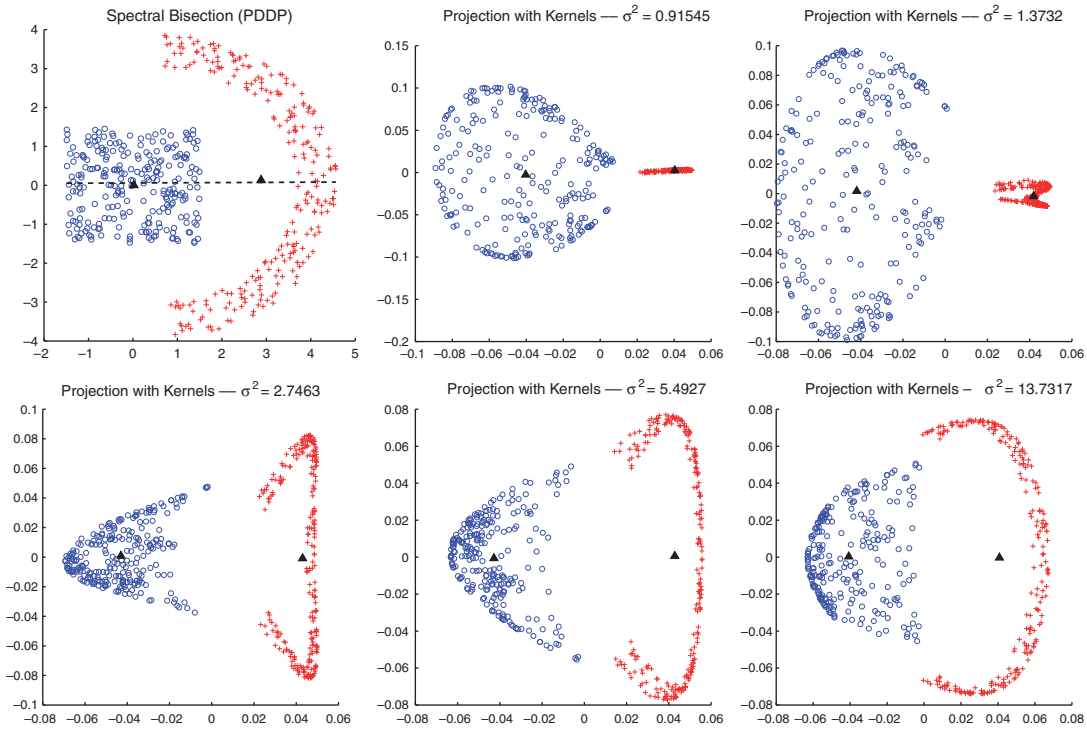

Figure 11. Original figure (top-left) and results of projections using kernels with different values of σ.

9. ILLUSTRATIVE EXAMPLES

The goal of this section is to demonstrate the behavior of the methods just seen on a few simple examples.

9.1. Projecting digits in 2-D space

Figure 12 shows the results of dimension reduction on a data set of handwritten digits ('0'–'9') [58] which consists of 200 samples per digit. Each sample was originally represented as a 649-dimensional feature vector, including the Fourier coefficients, profile correlations, Karhunen-Love coefficients, pixels averages, Zernike moments, and morphological features. Owing to the huge differences between the numeric ranges of the features, we normalize each feature such that the maximum value is one.

Here are the main observations from these plots. First, the supervised method LDA does well in separating the samples of different classes, as compared with the unsupervised method PCA. Both methods take into account the variances of the samples, but LDA makes a distinction between the 'within scatter' and 'between scatter', and outperforms PCA in separating the different classes. Second, both in theory and in practice, LLE and Eigenmaps share many similarities. For the present data set, both methods yield elongated and thin clusters. These clusters stretch out in the low-dimensional space, yet each one is localized and the different clusters are well separated. Our third observation concerns NPP and LPP, the linear variants of LLE and Eigenmaps, respectively. The methods should preserve the locality of each cluster just as their nonlinear counterparts. They yield bigger cluster shapes instead of the 'elongated and thin' ones of their nonlinear counterparts. The fourth observation is that ONPP and OLPP, the orthogonal variants of NPP and LPP, yield poorly separated projections of the data in this particular case. The samples of the same digit are distributed in a globular shape (possibly with outliers), but for different digits, the samples just mingle together, yielding a rather undesirable result. Although the orthogonal projection methods OLPP and ONPP do quite a good job for face recognition (see Section 9.3.2, and results in [16])

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

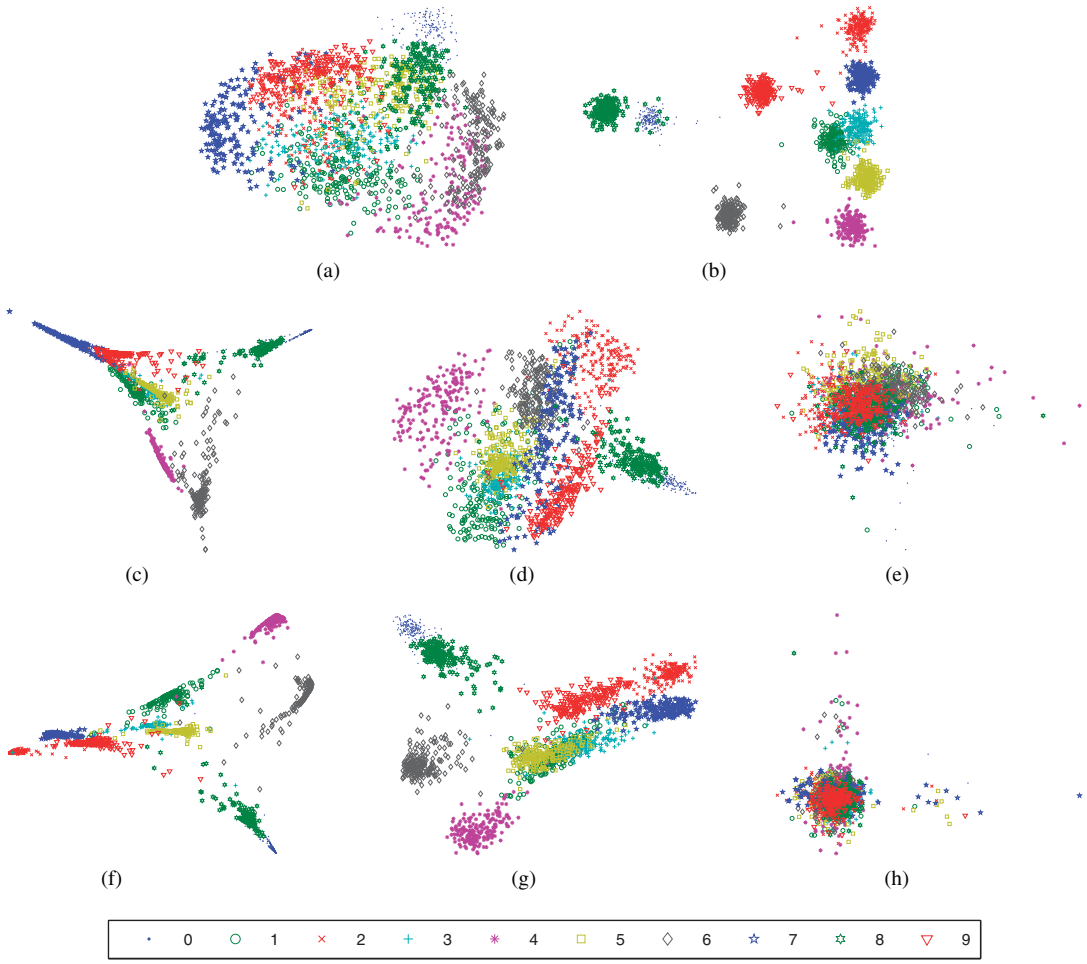

Figure 12. Low-dimensional (2D) representations of handwritten digits: (a) PCA; (b) LDA; (c) LLE; (d) NPP; (e) ONPP; (f) Eigenmaps; (g) LPP; and (h) OLPP.

they yield poor 2-D projections in this case. A possible explanation is that we are projecting data from a high-dimensional space to a space of dimension only two, whereas face recognition methods utilize much higher dimensions in order to successfully classify faces. The problem is also intrinsically different. In the current situation we are trying to visualize a clustering of many data items on a 2-D plane, whereas in classification we use the projected d-dimension data to compare a test image to other images, which are labeled. The visual clustering of the data when projected in 2-D space does not matter.

9.2. *Effect of kernelization*

We consider the same data set as in Section 9.1, but now fewer digits are taken for each experiment. Specifically, we look at digits that are usually more difficult to distinguish, and we select first the three digits '5', '8' and '9'. We consider only two methods here, namely PCA and OLPP, and their kernel versions, K-PCA and K-OLPP (Figures 13, 14).

For the kernel version we use the same Gaussian kernel $K(x, y) = \exp(-\|x-y\|_2^2/\sigma^2)$ as in Section 8.6. As suggested in Section 8.6, the parameter σ is selected to be half the median of all pairwise distances obtained from a random sample of 1000 points.[‡‡] This typically results in a reasonable estimate of the best σ.

[‡‡]If the data set contains fewer than 1000 samples then all samples are used.

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

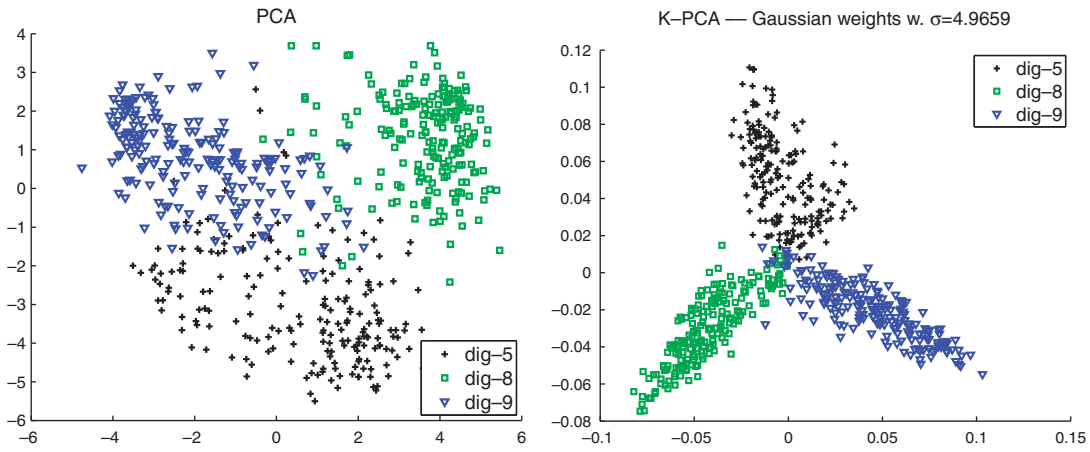

Figure 13. PCA and K-PCA for digits 5, 8, and 9 of data set mfeat.

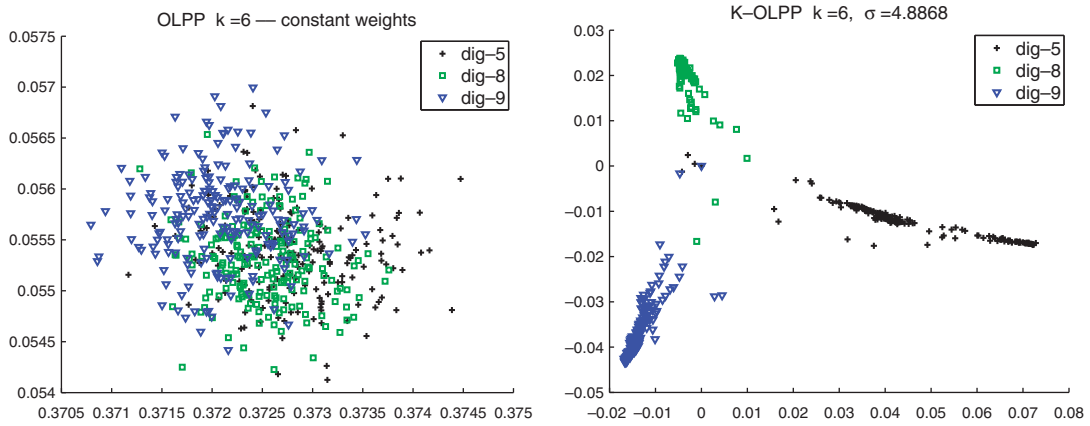

Figure 14. OLPP and K-OLPP for digits 5, 8, and 9 of data set mfeat.

The improvement seen from the standard versions to the kernel versions is striking. Clearly, not all values of σ will yield a good improvement. For example when we tried taking four digits, the results for basically any σ were rather poor for this particular data set.

The next test example uses another digit data set, one which is publicly available[7]. This data set contains 39 samples from each class (the digits '0'–'9'). Each digit image sample is represented lexicographically as a vector in space $\mathbb{R}^{320}$ and consists of zeros and ones. Figure 15 shows a random sample of 20 such pictures (20 pictures randomly selected out of the whole set of 390 pictures). As can be seen a few of the prints are rather difficult to decipher.

We repeat the previous experiment but this time we select four digits: 1, 3, 7, 9. The results are shown in Figures 16 and 17. The kernel used here is the same as before. As our set is not too large (156 images in all) we simply took σ to be equal to half the median of all pairwise distances in the set. The value of σ found in this way is shown in the corresponding plots.

The improvement seen from the standard versions to the kernel versions is remarkable. Just as before, not all values of σ will yield a good improvement.

9.3. Classification experiments

In this section we illustrate the methods discussed in the paper on two different classification tasks, namely, digit recognition and face recognition. Recall from Section 5 that the problem of classification is to determine the class of a test sample, given the class labels of previously seen

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

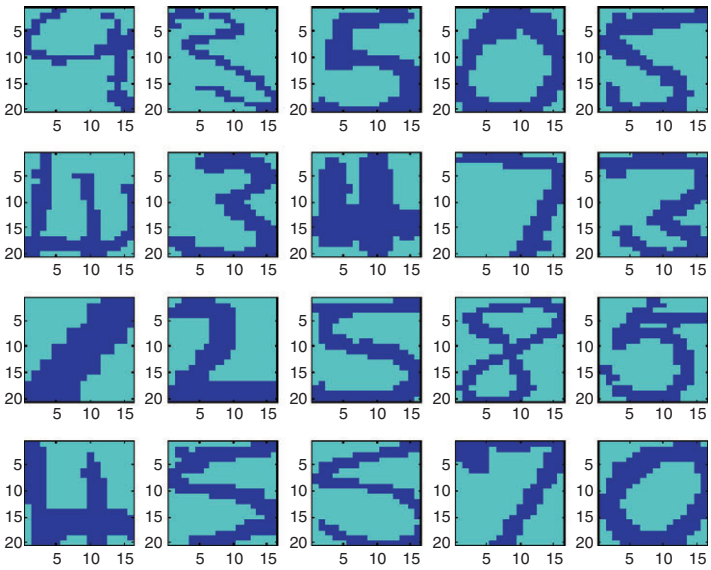

Figure 15. A sample of 20 digit images from the Roweis data set.

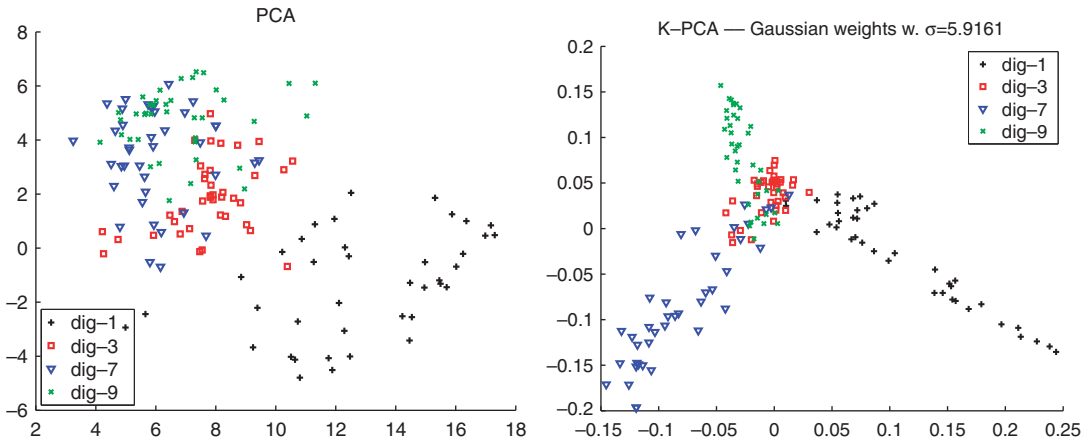

Figure 16. PCA and K-PCA for digits 1, 3, 7, 9 of the Roweis digits data set.

data samples (i.e. training data). Table II summarizes the characteristics of the data sets used in our evaluation. For digit recognition, we use the mfeat and Roweis data sets that were previously used in Sections 9.1 and 9.2. For face recognition, we use the UMIST [59], ORL [60] and AR [61] databases. We provide more information below.

- The UMIST database contains 20 people in different poses. The number of different views per subject varies from 19 to 48. We used a cropped version of the UMIST database that is publicly available [7]. Figure 18 illustrates a sample subject from the UMIST database along with its first 20 views.
- The ORL database contains 40 individuals and 10 different images for each individual including variation in facial expression (smiling/non-smiling) and pose. Figure 19 illustrates two sample subjects of the ORL database along with variations in facial expression and pose.
- The AR face database contains 126 individuals and 8 different images for each individual including variations in facial expression and lighting. Figure 20 illustrates two sample subjects of the AR database along with variations in facial expression and lighting.

In all graph-based methods we use supervised graphs, see Section 5.1. In the LPP and OLPP methods we use Gaussian weights, see Sections 8.6 and 9.2. The parameter σ is determined as

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

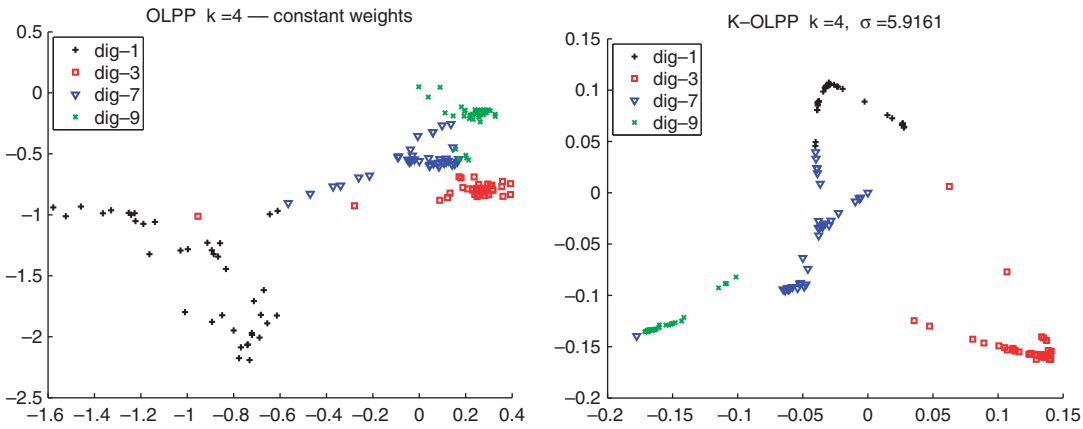

Figure 17. OLPP and K-OLPP for digits 1, 3, 7, 9 of the Roweis digits data set.

Table II. Data sets and their characteristics.

Data set	No. of classes	No. of samples per class
mfeat	10	200
Roweis	10	39
UMIST	20	19–48
ORL	40	10
AR	126	8

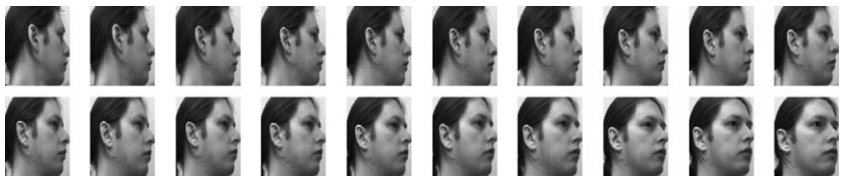

Figure 18. Sample from the UMIST database.

Figure 19. Sample from the ORL database.

described in Section 9.2. Finally, we should mention that the above methods have been pre-processed with a preliminary PCA projection step. The PCA projection is used in order to reduce the dimension of the data vectors to $n_{\text{train}} - c$, where n_{train} is the number of training samples (see e.g. [16, 29]). In what follows we discuss first recognition of handwritten digits and then face recognition. In both tasks, recognition is done in the reduced space, after dimension reduction, using nearest neighbor classification.

9.3.1. Handwritten digit recognition. This problem is of great practical importance to postal and delivery services around the world. The number of classes here is $c = 10$. We compare the linear

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

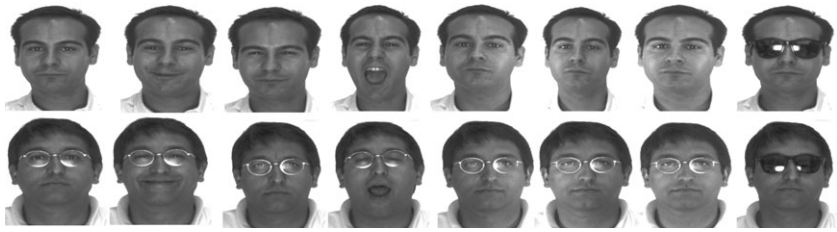

Figure 20. Sample from the AR database.

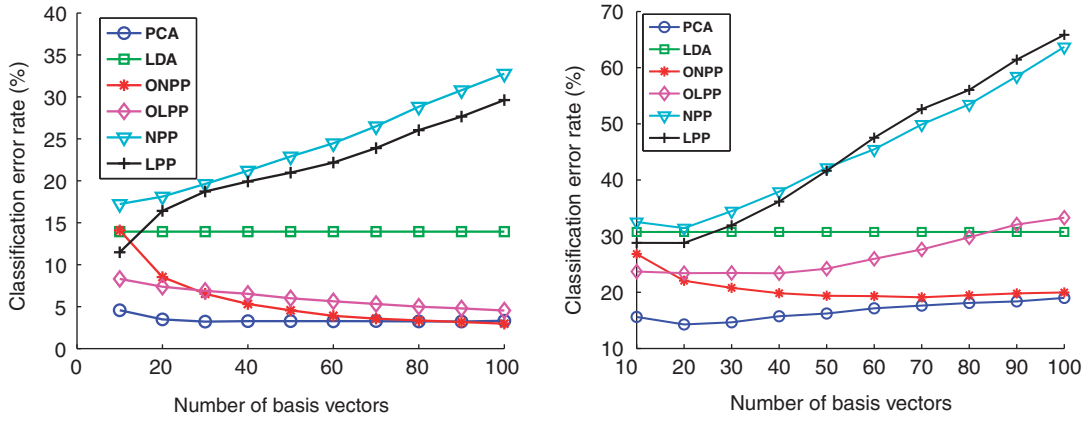

Figure 21. Handwritten digit recognition. Left panel: mfeat data set and right panel: Roweis data set.

dimension reduction methods discussed in this paper. We use 50 and 15 training samples per class in the mfeat and Roweis data sets respectively. The remainder of samples are assigned to the test set.

Figure 21 shows the average classification error rate of all methods with respect to dimension d of the reduced space. The averages are computed over 100 random formations of the training and test sets. Note that for LDA we only report the average performance at $d = c - 1$, as it cannot provide more than $c - 1$ discriminant axes.

First, observe that the performance of LPP parallels that of NPP. This is mostly due to Proposition 7.2, although in this case the relation $W = \hat{W} = H$ is not exactly true, due to the different weights used in each method (i.e. Gaussian weights in LPP and LLE weights in NPP). Second, notice that the orthogonal methods i.e. PCA, ONPP and OLPP offer the best performances and significantly outperform the non-orthogonal ones.

9.3.2. Face recognition. The problem of face recognition is somewhat similar to the one described for digit recognition. We want now to recognize subjects based on facial images. Face recognition has numerous applications such as surveillance, automated screening, authentication and human-computer interaction, to name just a few.

We use 5, 10 and 5 training samples per class in the ORL, UMIST, and AR data sets respectively, while the remainder of samples are assigned to the test set. Figures 22 and 23 show the average classification error rates of all methods on the above three data sets. The averages are computed over 100 random formations of the training and test sets. As was previously done, for LDA we only report the average performances up to $d = c - 1$. Notice again that the orthogonal methods are in general superior to the non-orthogonal ones. Observe also that the orthogonal graph-based methods, ONPP and OLPP, are the best performers for the face recognition task.

One reason that orthogonal projection methods do well for classification may be that distances are not too distorted when projecting data. Indeed $\|V^{\mathrm{T}}(x - y)\| \leqslant \|x - y\|$, and in fact this distance

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

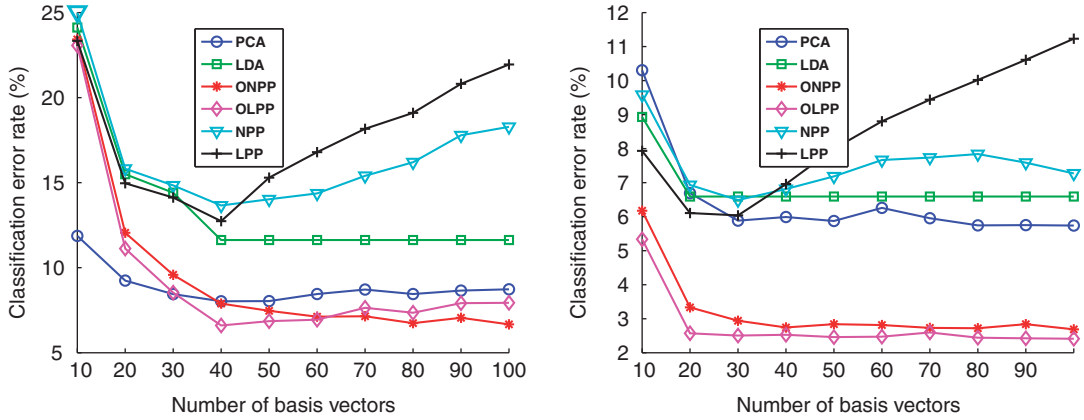

Figure 22. Face recognition results on the ORL (left) and UMIST (right) data sets.

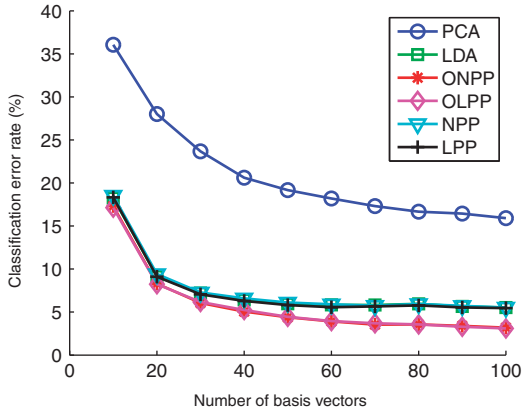

Figure 23. Face recognition results on the AR data set.

may be fairly accurate for points belonging to X due to the choice of V (e.g. when columns of V consist of the singular vectors of X as in PCA).

10. BEYOND SPECTRAL METHODS AND TRACE OPTIMIZATION

While this paper focused on dimension reduction based on spectral techniques and trace optimization, other existing powerful methods rely on convex optimization with constraints. This section briefly describes two examples in this class for illustration purposes. For a recent survey of these techniques see [62].

Possibly the best known technique along these lines in supervised learning is the method of Support Vector Machines (SVM); see [8, 49, 63]. It is in spirit similar to LDA (cf. Section 5.2) whereas it finds a one-dimensional projection to separate the data in some optimal way. Formally, the SVM approach consists of finding a hyperplane which best separates two training sets belonging to two classes. If the hyperplane is $w^{\mathrm{T}}x+b=0$, then the classification function would be $f(x)=\mathrm{sign}(w^{\mathrm{T}}x+b)$. This will assign the value $y=+1$ to one class and $y=-1$ to the other, and it is capable of perfectly separating the two classes in ideal situations when the classes are linearly separable.

One of the key ingredients used by SVM is the notion of *margin*, which is the distance between two parallel support planes for the two classes. First, observe that the parameters w, b can be

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

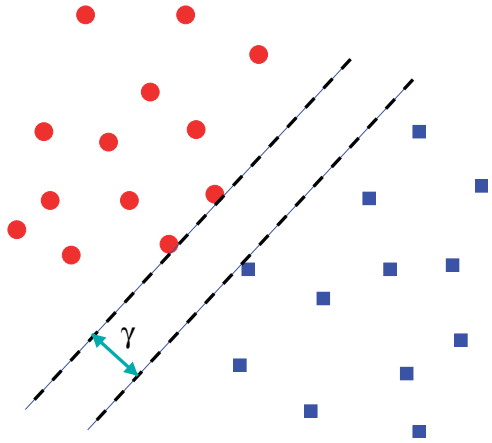

Figure 24. Illustration of the margin in SVM.

normalized by looking for hyperplanes of the form $w^{\mathrm{T}}x+b\geqslant 1$ to include one set and $w^{\mathrm{T}}x+b\leqslant -1$ to include the other. With $y_i=+1$ for one class and $y_i=-1$ for the other, we can write the constraints as $y_i(w^{\mathrm{T}}x_i+b)\geqslant 1$. The margin is the maximum distance between two such planes. SVM finds w, b so that the margin is maximized.

Therefore, SVM finds the best separating hyperplane (middle of the two support planes) by maximizing the margin subjected to the constraint $y_i(w^{\mathrm{T}}x_i+b)\geqslant 1$. As it turns out the margin is given by $\gamma=2/\|w\|_2$. (Figure 24 shows an illustration.) This leads to the following constrained quadratic programming problem:

$$\begin{aligned} &\min_{w.b} \quad \tfrac{1}{2}\|w\|_2^2, \\ &\text{s.t.} \quad y_i(w^{\mathrm{T}}x_i+b)\geqslant 1 \quad \forall x_i. \end{aligned}$$

Often the dual problem is solved instead of the above primal problem. In case the two classes are not separable, the constraint is relaxed by introducing slack variables. In addition, the problem is often solved in 'feature space', meaning simply that a kernel is used to redefine the inner product to enable a linear separation of the two classes.

There are several other types of optimization problems involving Semi-Definite Programming, in which the optimization problem involves matrices that are constrained to be semi-positive definite. *Maximum Variance Unfolding* (MVU) is one such example; see [64, 65]. Assume we have a certain affinity graph available. We could wish to find a set of centered samples in low-dimensional space (constraint: $\sum_i y_i=0$) which maximize the variance $\sum_i \|y_i\|_2^2$ with the constraint that $\|y_i-y_j\|_2=\|x_i-x_j\|_2$ whenever (x_i, x_j) are linked by an edge. This is a quadratic programming problem with quadratic constraints. It is possible to provide a solution in terms of the matrix Grammian of the low-dimensional data, i.e. $K=Y^{\mathrm{T}}Y$. This then leads to the following semi-definite program:

$$\text{Maximize} \sum_i K_{ii} \quad \text{s.t.} \begin{cases} \text{(i)} & K_{ii}+K_{jj}-2K_{ij}=\|x_i-x_j\|_2^2 \text{ if } (x_j,x_j)\in E, \\ \text{(ii)} & \sum_{ij} K_{ij}=0, \\ \text{(iii)} & K\succ 0. \end{cases}$$

Once the matrix K is found, one computes Y of dimension $d\times n$ such $Y^{\mathrm{T}}Y=K$ and this involves a diagonalization of K.

We have given just two examples (one supervised, one unsupervised) of methods involving more complex techniques (i.e. optimization) than those methods seen in earlier sections, which were based on (projected) eigenvalue problems. Many other convex optimization formulations have been discussed in, e.g. [66–68]. We point out that these optimization methods tend to be far more

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

expensive than spectral methods and this limits their capability for handling large-scale problems. For this reason, simpler techniques resorting to spectral problems are sometimes preferred. Realistic large-scale systems can have millions or even billions of variables and constraints and this puts them out of the reach of methods based on these sophisticated optimization techniques. A common alternative in such situations is to perform sampling on the data and reduce the problem size. This is the case for MVU, where a landmark version [69] was proposed if the sample size becomes large. Yet another alternative is to apply heuristics and/or to relax the constraints in order to find approximate solutions. In contrast, as long as the matrix is sparse, eigenvalue problems can still be efficiently solved.

11. CONCLUSION

This paper gave an overview of spectral problems that arise in dimension reduction methods, with an emphasis on the many interrelations between the various approaches used in the literature. These dimension reduction methods are often governed by a trace optimization problem with constraints, along with some data locality criteria. When viewed from this angle, and with the help of kernels, one can easily define a comprehensive unifying framework for dimension reduction methods. The illustrative examples shown indicate that in spite of their seemingly similar nature, these methods can perform very differently for a given task.

Many challenging issues remain for a linear algebra specialist interested in this topic to explore. For example, although kernels are indeed very powerful, we do not know how to select them (optimally) for a specific data set and problem. Moreover, kernel methods lead to large $n \times n$ matrices, typically dense, which are difficult to handle in practice. This leads to a broader issue that remains a problem in this area, namely the general question of computational cost. Methods considered in the literature so far have often relied on very expensive matrix factorizations, the most common being the SVD. In view of the ever-increasing sizes of practical data sets, it has now become critical to search for less costly alternatives.

ACKNOWLEDGEMENTS

The authors thank the referees for their helpful remarks that helped to improve this paper. While finalizing the writing of the revised version of this paper, we heard the sad and unexpected news of the passing away of Sam Roweis whose name is mentioned several times in the paper. Sam Roweis' generous posting of articles, lecture notes, matlab scripts, and data, had a major impact on our own understanding of many of the methods discussed in this paper.

REFERENCES

1. Webb A. *Statistical Pattern Recognition* (2nd edn). Wiley: Hoboken, NJ, 2002.
2. Koren Y. On spectral graph drawing. *COCOON 03*, LNCS, vol. 2697. Springer: Berlin, 2003; 496–508.
3. Noack A. An energy model for visual graph clustering. *Proceedings of the 11th International Symposium on Graph Drawing* (*GD 2003*), *LNCS 2912*. Springer: Berlin, 2004; 425–436.
4. Curtarolo S, Morgan D, Persson K, Rodgers J, Ceder G. Predicting crystal structures with data mining of quantum calculations. *Physical Review Letters* 2003; **91**(13):135503.
5. Ceder G, Morgan D, Fischer C, Tibbetts K, Curtarolo S. Data-mining-driven quantum mechanics for the prediction of structure. *MRS Bulletin* 2006; **31**:981–985.
6. Sebastian SE, Harrison N, Batista CD, Balicas L, Jaime M, Sharma PA, Kawashima N, Fisher IR. Dimensional reduction at a quantum critical point. *Nature* 2006; **441**:617.
7. http://www.cs.toronto.edu/∼roweis/data.html.
8. Bishop CM. Pattern Recognition and Machine Learning. *Information Science and Statistics*. Springer: Berlin, 2006.
9. Roweis S, Saul L. Nonlinear dimensionality reduction by locally linear embedding. *Science* 2000; **290**:2323–2326.
10. John Lee A, Verleysen M. *Nonlinear Dimensionality Reduction. Information Science and Statistics*. Springer: Berlin, 2007.
11. Belkin M, Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* 2003; **15**(6):1373–1396.

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

12. Zhang Z, Zha H. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal on Scientific Computing* 2005; **26**(1):313–338.
13. Tenenbaum JB, de Silva V, Langford JC. A global geometric framework for nonlinear dimensionality reduction. *Science* 2000; **290**:2319–2323.
14. Weinberger KQ, Saul LK. Unsupervised learning of image manifolds by semidefinite programming. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (*CVPR*), Washington, DC, U.S.A., 2004.
15. Sha F, Saul LK. Analysis and extension of spectral methods for nonlinear dimensionality reduction. *Proceedings of the 22nd International Conference on Machine Learning* (*ICML*), Bonn, Germany, 2005.
16. Kokiopoulou E, Saad Y. Orthogonal neighborhood preserving projections. In *IEEE 5th International Conference on Data Mining* (*ICDM05*), Houston, TX, 27–30 November Han J *et al.* (eds). IEEE: New York, 2005; 234–241.
17. Ham J, Lee DD, Mika S, Schölkopf B. A kernel view of the dimensionality reduction of manifolds. *ICML '04*: *Proceedings of the 21st International Conference on Machine Learning*. ACM: New York, NY, U.S.A., 2004; 47.
18. Williams CKI. On a connection between kernel PCA and metric multidimensional scaling. *Machine Learning* 2002; **46**(1–3):11–19.
19. Parlett BN. *The Symmetric Eigenvalue Problem. Number 20 in Classics in Applied Mathematics*. SIAM: Philadelphia, 1998.
20. Saad Y. *Numerical Methods for Large Eigenvalue Problems*. Halstead Press: New York, 1992.
21. Guo Y-F, Li S-J, Yang J-Y, Shu T-T, Wu L-D. A generalized Foley–Sammon transform based on generalized Fisher discriminant criterion and its application to face recognition. *Pattern Recognition Letters* 2003; **24**(1–3):147–158.
22. Wang H, Yan SC, Xu D, Tang XO, Huang T. Trace ratio vs. ratio trace for dimensionality reduction. *IEEE Conference on Computer Vision and Pattern Recognition*, 2007; 17–22.
23. Xiang S, Nie F, Zhang C. Learning a mahalanobis distance metric for data clustering and classification. *Pattern Recognition* 2008; **41**(12):3600–3612.
24. Yan S, Tang X. Trace quotient problems revisited. In *Proceedings of the European Conference on Computer Vision* (*Lecture Notes in Computer Science*, vol. 2), Leonardis A, Bischof H, Pinz A (eds). Springer: Berlin–Heidelberg, 2006; 232–244, Number 3952.
25. Nie F, Xiang S, Jia Y, Zhang C. Semi-supervised orthogonal discriminant analysis via label propagation. *Pattern Recognition* 2009; **42**:2615–2627.
26. Saul L, Roweis S. Think globally, fit locally: unsupervised learning of nonlinear manifolds. *Journal of Machine Learning Research* 2003; **4**:119–155.
27. Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems 14*. MIT Press: Cambridge, MA, 2001; 585–591.
28. He X, Niyogi P. Locality preserving projections. *Proceedings of Conference on Advances in Neural Information Processing Systems*, Vancouver, BC, Canada, 2003.
29. Kokiopoulou E, Saad Y. Orthogonal neighborhood preserving projections: a projection-based dimensionality reduction technique. *IEEE TPAMI* 2007; **29**:2143–2156.
30. Jolliffe IT. *Principal Component Analysis*. Springer: New York, 1986.
31. Torgerson WS. Multidimensional scaling: I. Theory and method. *Psychometrika* 1952; **17**(4):401–419.
32. Cai D, He X, Han J, Zhang H-J. Orthogonal Laplacian faces for face recognition. *IEEE Transactions on Image Processing* 2006; **15**(11):3608–3614.
33. Howland P, Park H. Generalizing discriminant analysis using the generalized singular value decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2004; **26**(8):995–1006.
34. Zhang D, Zhou Z-H, Chen S. Semi-supervised dimensionality reduction. *SIAM Data Mining*, 2007; 629–634.
35. Cai D, He X, Han J. Semi-supervised discriminant analysis. *IEEE International Conference on Computer Vision* (*ICCV*), Rio de Janeiro, Brazil, 2007.
36. Song Y, Nie F, Zhang C, Xiang S. A unified framework for semi-supervised dimensionality reduction. *Pattern Recognition* 2008; **41**:2789–2799.
37. Zhang Y, Yeung D-Y. Semi-supervised discriminant analysis using robust path-based similarity. *IEEE International Conference on Computer Vision and Pattern Recognition* (*CVPR*), Anchorage, Alaska, 2008.
38. Zhang Y, Yeung D-Y. Semi-supervised discriminant analysis via CCCP. *ECML/PKDD*, 2008; 644–659.
39. Yang X, Fu H, Zha H, Barlow J. Semi-supervised nonlinear dimensionality reduction. *The 23rd International Conference on Machine Learning* (*ICML*), Pittsburg, PA, U.S.A., 2006.
40. Zhang Z, Zha H, Zhang M. Spectral methods for semi-supervised manifold learning. *IEEE International Conference on Computer Vision and Pattern Recognition* (*CVPR*), Anchorage, Alaska, 2008.
41. Shi J, Malik J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis Machine Intelligence* 2000; **22**(8):888–905.
42. Ng AY, Jordan M, Weiss Y. On spectral clustering: analysis and an algorithm. *Advances in Neural Information Processing Systems 14*, 2002.
43. Ding C. Spectral clustering. *International Conference on Machine Learning 2004 Tutorial*, 2004.
44. Luxburg U. A tutorial on spectral clustering. *Statistics and Computing* 2007; **17**(4):395–416.
45. Hagen L, Kahng AB. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits Systems* 1992; **11**(9):1074–1085.
46. Fiedler M. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal* 1973; **23**:298–305.

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla

47. Fiedler M. A property of eigenvectors of nonnegative symmetric matrices and its applications to graph theory. *Czechoslovak Mathematical Journal* 1975; **25**:619–633.
48. Müller KR, Mika S, Ratsch G, Tsuda K, Scholkopf B. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks* 2001; **12**:181–201.
49. Vapnik V. *Statistical Learning Theory*. Wiley: New York, 1998.
50. Schlköpf B, Smola AJ. *Learning with Kernels*: *Support Vector Machines*, *Regularization*, *Optimization and Beyond*. MIT Press: Cambridge, MA, 2001.
51. Shawe-Taylor J, Cristianini N. *Kernel Methods for Pattern Analysis*. Cambridge University Press: Cambridge, 2004.
52. Aronszajn N. Theory of reproducing kernels. *Transactions of the American Mathematical Society* 1950; **68**: 337–404.
53. Schölkopf B, Smola A, Muller K. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* 1998; **10**:1299–1319.
54. Fouss F, Pirotte A, Renders J-M, Saerens M. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering* 2007; **19**(3):355–369.
55. Bengio Y, Paiement J-F, Vincent P, Delalleau O, Le Roux N, Ouimet M. Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and spectral clustering. In *Advances in Neural Information Processing Systems 16*, Vancouver, Canada, Thrun S, Saul L, Schölkopf B (eds). MIT Press: Cambridge, MA, 2004.
56. Boley D. Principal direction divisive partitioning. *Data Mining and Knowledge Discovery* 1998; **2**(4):325–344.
57. Schölkopf B, Smola A. *Learning with Kernels*. MIT press: Cambridge, MA, 2002.
58. Asuncion A, Newman DJ. UCI machine learning repository (multiple features data set). Available from: http://www.ics.uci.edu/~mlearn/MLRepository.html.
59. Graham DB, Allinson NM. Characterizing virtual eigensignatures for general purpose face recognition. *Face Recognition*: *From Theory to Applications* 1998; **163**:446–456.
60. Samaria F, Harter A. Parameterisation of a stochastic model for human face identification. *The 2nd IEEE Workshop on Applications of Computer Vision*, Sarasota FL, December 1994.
61. Martinez AM, Benavente R. The AR face database. *Technical Report 24*, CVC, 1998.
62. Bennett KP, Parrado-Hernandez E. The interplay of optimization and machine learning research. *Journal of Machine Learning Research* 2006; **7**:1265–1281.
63. Cortes C, Vapnik V. Support–vector networks. *Machine Learning* 1995; **20**(3):273–297.
64. Weinberger KQ, Saul LK. Unsupervised learning of image manifolds by semidefinite programming. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR-04*), 2004; **2**:988–995.
65. Weinberger KQ, Saul LK. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. *AAAI'06*: *Proceedings of the 21st National Conference on Artificial Intelligence*. AAAI Press, 2006; 1683–1686.
66. Xu L, Neufeld J, Larson B, Schuurmans D. Maximum margin clustering. *Advances in Neural Information Processing Systems 17*, 2005.
67. Xu L, Schuurmans D. Unsupervised and semi-supervised multi-class support vector machines. *Proceedings of the 20th National Conference on Artificial Intelligence*, Pittsburg, PA, U.S.A., 2005.
68. Bach F, Harchaoui Z. Diffrac: a discriminative and flexible framework for clustering. *Advances in Neural Information Processing Systems 20*, 2008.
69. Weinberger K, Packer B, Saul L. Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization. *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, Barbados, 2005.

Copyright © 2010 John Wiley & Sons, Ltd.

DOI: 10.1002/nla