Number PH-2002-01
in the Pattern Recognition Group Technical Report Series
*(submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence)*

# Locally linear embedding for classification

Dick de Ridder and Robert P.W. Duin

dick@ph.tn.tudelft.nl
http://www.ph.tn.tudelft.nl/~dick

Delft University of Technology

Pattern Recognition Group
Imaging Science & Technology Department
Faculty of Applied Science
Delft University of Technology
Lorentzweg 1, 2628 CJ Delft, The Netherlands
telephone: +31 15 2781416 fax: +31 15 2786740
http://www.ph.tn.tudelft.nl/

## Abstract

Locally linear embedding (LLE) is a recently proposed unsupervised procedure for mapping high-dimensional data nonlinearly to a lower-dimensional space. In this paper, a supervised variation on LLE is proposed. This mapping, when combined with simple classifiers such as the nearest mean classifier, is shown to yield remarkably good classification results in experiments. Furthermore, a number of algorithmic improvements are proposed which should ease application of both traditional and supervised LLE by eliminating the need for setting some of the parameters.

## Keywords

nonlinear mapping, supervised mapping, regularisation, intrinsic dimensionality

## Contents

## 1 Introduction

In many pattern recognition problems, sensors provide a large amount of measurements (features), such as in images and sound signals. Applying unsupervised techniques (e.g. clustering methods) or supervised techniques (e.g. classifiers) directly to these measurements, i.e. pixels or samples, is problematic due to the parameter estimation problems inherent in applying learning methods to high-dimensional data sets with a limited number of samples. Before such techniques can be applied with a reasonable hope of generalisation, a small number of useful features will have to be extracted. That is, the dimensionality of the feature space will have to be reduced.

Traditional methods to perform dimensionality reduction are mainly linear, and include selection of subsets of measurements and linear mappings to lower-dimensional spaces [6,7,11]. Over the years, a number of techniques have been proposed to perform nonlinear mappings, such as multi-dimensional scaling [2], the self-organising map [12] and generative topographic mapping [1], principal curves and surfaces [9], auto-encoder neural networks [5] and mixtures of linear models [16]. All of these are problematic in application in some way: multi-dimensional scaling and neural networks are hard to train and time-consuming, as are principal curves and surfaces. The latter, as well as the generative topographic mapping, need large data sets to estimate their many parameters. Mixtures of localised linear models require the user to set a number of parameters, which are highly specific to each data set and determine how well the model fits the data.

Recently, a conceptually simple yet powerful method for nonlinear mapping has been proposed by Saul and Roweis [15]: locally linear embedding (LLE). The basic idea is that of global minimisation of the reconstruction error of the set of all local neighbourhoods in the data set. Although the authors demonstrate their algorithm on a number of artificial and realistic data sets, there have as yet been few reports of application of LLE. Problems one might encounter are that there are a number of parameters to be set, which greatly influence the results obtained, and a high computational demand. In this paper, some of these problems will be addressed. Next, it will be shown how class label information can be used in a supervised application of LLE. Experiments will show the combination of supervised LLE and simple classifiers to give very good performance.

The remainder of this paper is laid out as follows. In section 2, LLE will be reviewed. Next, some elements of the algorithm will be looked at in more depth and some proposals are made to facilitate practical use of LLE. In section 4 a variation on LLE for supervised problems will be discussed and some experiments will be presented. Finally, section 5 ends with some conclusions.

## 2 Locally linear embedding

LLE [15] maps a data set $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, $\mathbf{x}_i \in \mathbb{R}^d$ globally to a data set $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$, $\mathbf{y}_i \in \mathbb{R}^m$. Assuming the data lies on a nonlinear manifold which locally can be approximated linearly, it uses two stages: (I) locally fitting hyperplanes around each sample $\mathbf{x}_i$, based on its $k$ nearest neighbours, and calculating reconstruction weights, and (II) finding lower-dimensional co-ordinates $\mathbf{y}_i$ for each $\mathbf{x}_i$, by minimising a mapping function based on these weights.

In stage $\mathrm{I}$, the cost function minimised is:

$$\mathcal{E}_I(W) = \sum_{i=1}^{n} \left| \mathbf{x}_i - \sum_{j=1}^{k} w_j^{(i)} \mathbf{x}_{N(j)} \right|^2 , \tag{1}$$

i.e. how well each $\mathbf{x}_i$ can be linearly reconstructed in terms of its neighbours $\mathbf{x}_{N(1)} \ldots \mathbf{x}_{N(k)}$. For one vector $\mathbf{x}_i$ and weights $w_j^{(i)}$ that sum up to 1, this gives a contribution

$$\begin{aligned} \mathcal{E}_I^{(i)}(W) &= \left| \sum_{j=1}^{k} w_j^{(i)} (\mathbf{x}_i - \mathbf{x}_{N(j)}) \right|^2 \\ &= \sum_{j=1}^{k} \sum_{m=1}^{k} w_j^{(i)} w_m^{(i)} Q_{jm}^{(i)}, \end{aligned} \tag{2}$$

where $\mathbf{Q}^{(i)}$ is the $k \times k$ matrix

$$Q_{jm}^{(i)} = (\mathbf{x}_i - \mathbf{x}_{N(j)})^T (\mathbf{x}_i - \mathbf{x}_{N(m)}) \tag{3}$$

Let $\mathbf{R}^{(i)} = (\mathbf{Q}^{(i)})^{-1}$. Solving the least squares problem (eqn. 2) with constraint $\sum_j w_j^{(i)} = 1$ gives:

$$w_j^{(i)} = \frac{\sum_{m=1}^{k} R_{jm}^{(i)}}{\sum_{p=1}^{k} \sum_{q=1}^{k} R_{pq}^{(i)}}. \tag{4}$$

In practice, a regularisation parameter $r$ will have to be used for $\mathbf{Q}^{(i)}$ before inversion (as its rank is $d$, certainly for $k > d$): $\mathbf{R}^{(i)} = (\mathbf{Q}^{(i)} + r\mathbf{I})^{-1}$.

Interestingly, $\mathbf{Q}^{(i)}$ can also be calculated based on just the *squared* Euclidean distance matrix $\mathbf{D}$ between all samples in $\mathcal{X}$:

$$Q_{jm}^{(i)} = \frac{1}{2} \left( D_{i,N(j)} + D_{i,N(m)} - D_{N(j),N(m)} \right). \tag{5}$$

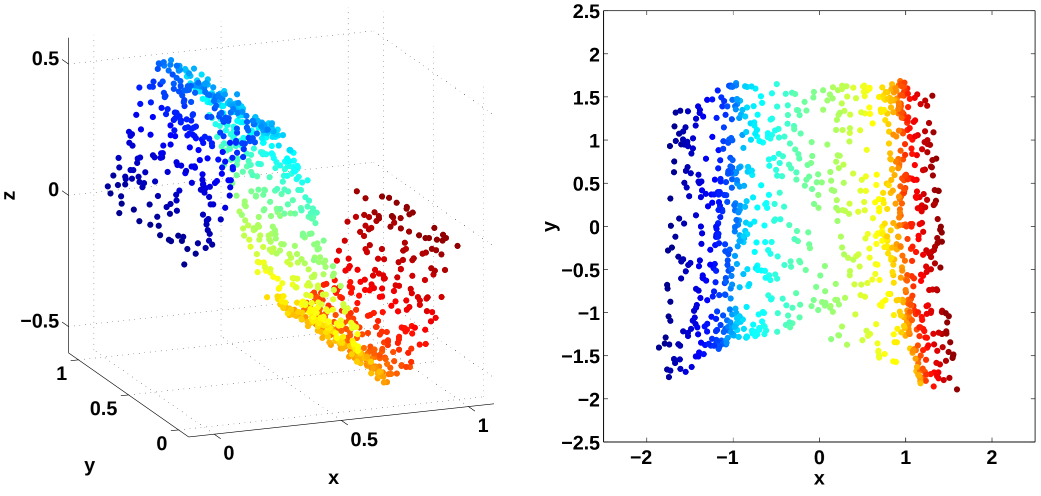
In stage $\mathrm{II}$, the weights $w$ are fixed and new $m$-dimensional vectors $\mathbf{y}_i$ are sought which minimise the criterion:

$$\mathcal{E}_{II}(\mathbf{Y}) = \sum_{i=1}^{n} \left| \mathbf{y}_i - \sum_{j=1}^{k} w_j^{(i)} \mathbf{y}_{N(j)} \right|^2 . \tag{6}$$

The $w_j^{(i)}$'s can be stored in an $n \times n$ sparse matrix $\mathbf{W}$, where $W_{i,N(j)} = w_j^{(i)}$. Re-writing eqn. 6 then gives

$$\mathcal{E}_{II}(\mathbf{Y}) = \sum_{i=1}^{n} \sum_{j=1}^{n} M_{ij} \mathbf{y}_i^T \mathbf{y}_j = \mathrm{tr}(\mathbf{Y}\mathbf{M}\mathbf{Y}^T), \tag{7}$$

where $\mathbf{M}$ is an $n \times n$ matrix found as $\mathbf{M} = (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W})$, and $\mathbf{Y}$ contains the $\mathbf{y}_i$'s as its columns.



(a)

(b)

**Fig. 1.** (a) $\mathcal{X}$ ($d = 3$, $n = 1000$): data sampled from an sine-shaped manifold, corrupted by additive Gaussian noise ($\sigma = 0.01$). (b) LLE-mapped $\mathbf{Y}$ ($m = 2$, $k = 15$, $r = 3 \times 10^{-3}$).

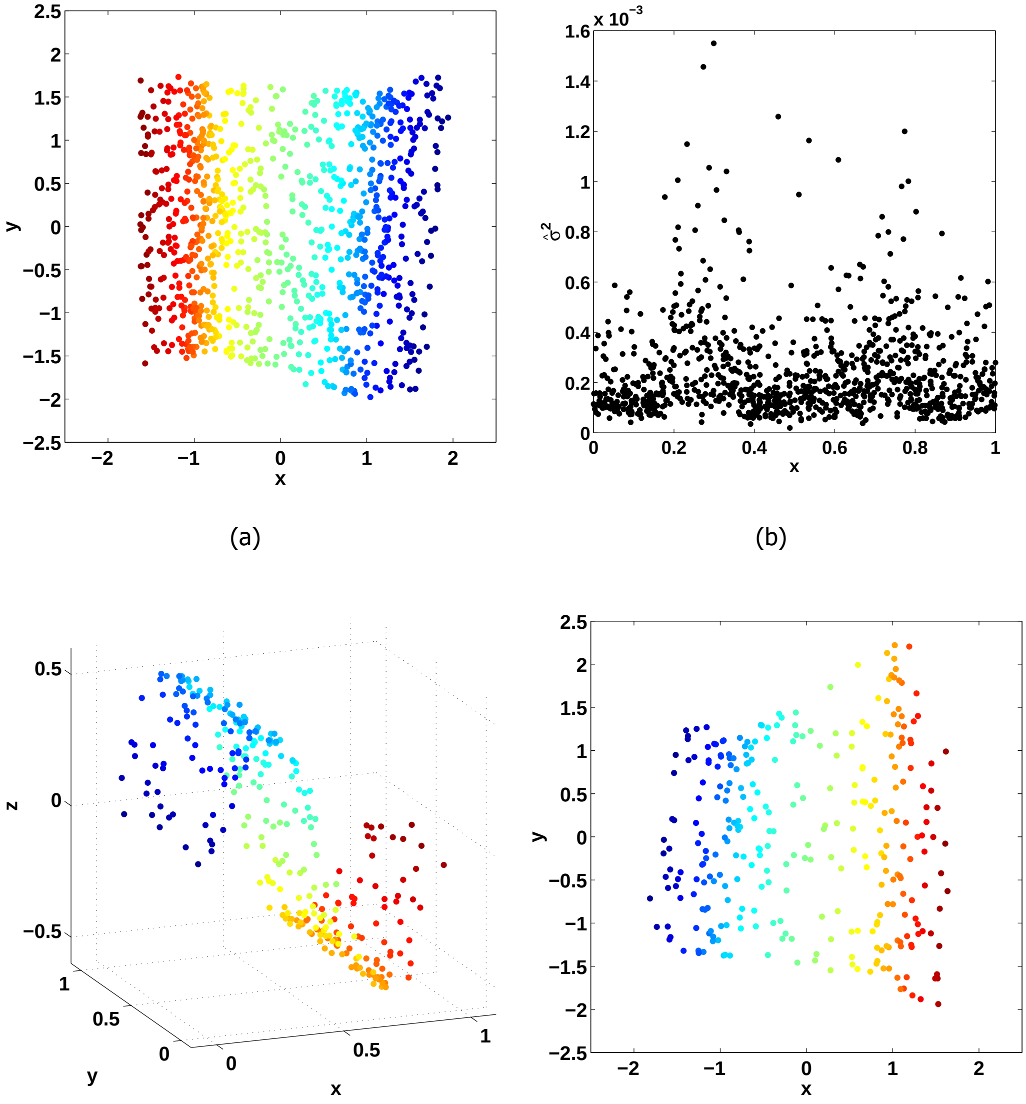
To be able to solve this problem, the covariance matrix of the $\mathbf{y}$'s can be constrained to be identity (otherwise $\mathbf{y}_i = \mathbf{0}, \forall i$ would be optimal). Finding $\mathbf{Y}$ then becomes a well-known problem: minimise tr$(\mathbf{YMY}^T)$ with constraint $\frac{1}{n}\mathbf{YY}^T = \mathbf{I}$. Using Lagrange multipliers and setting the derivative to zero gives $(\mathbf{M} - \mathbf{\Lambda})\mathbf{Y}^T = 0$, where $\mathbf{\Lambda}$ is the diagonal Lagrange multiplier matrix. Clearly, this is an eigenvalue problem. All eigenvectors of $\mathbf{M}$ are solutions, but the eigenvectors corresponding to the smallest eigenvalues minimise $\mathcal{E}_{II}(\mathbf{Y})$ [8]. The eigenvector with the smallest eigenvalue corresponds to the mean of $\mathbf{Y}$ and can be discarded to enforce $\sum_{i=1}^{n} \mathbf{y}_i = \mathbf{0}$. The next $m$ eigenvectors then give the $\mathbf{Y}$ that minimises eqn. 7.

As an example, figures 1(a) and (b) shows a 3D data set LLE-mapped to 2D. The choice of $r$ is rather important; for $r = 0$ the eigenanalysis (MATLAB's `eigs` routine) does not converge.

A new, previously unseen sample $\mathbf{x}$ can be mapped by finding its $k$ nearest neighbours in $\mathcal{X}$, calculating new reconstruction weights $w_j$ according to eqn. 4 and mapping it using $\mathbf{y} = \sum_{j=1}^{k} w_j \mathbf{y}_{N(j)}$. This is not a parametric mapping; still, the axes of the mapped space can be seen as corresponding to latent variables in the original data.

## 3 Improvements to the algorithm

The previous section showed that to find a good LLE mapping, three parameters will have to be set: the dimensionality $m$ to map to, the number of neighbours $k$ to take into account and the regularisation parameter $r$. Mapping quality is quite sensitive to these parameters. If $m$ is set too high, the mapping will enhance noise (due to the constraint $\frac{1}{n}\mathbf{YY}^T = \mathbf{I}$); if it is set too low, distinct parts of the data set might be mapped on top



(a)

(b)

(c)

(d)

**Fig. 2.** (a) LLE-mapped $\mathbf{Y}$ ($v = 0.90$, $k = 15$, $r$ found automatically). (b) $\hat{\sigma}_i^2$ as a function of the $x$-axis of figure 1(a). (c) Selected subset ($p = 300$). (d) LLE-mapped subset $\mathbf{Y}$ ($v = 0.90$, $k = 15$, $r$ found automatically).

of each other. If $k$ is set too small, the mapping will not reflect any global properties; if it is too high, the mapping will lose its nonlinear character and behave like traditional PCA, as the entire data set is seen as local neighbourhood. Finally, if $r$ is set incorrectly, the eigenanalysis may not converge.

Before we discuss how LLE can be used for classification, in section 4, some methods for easier setting of these parameters will be discussed. Most of these are based on the following observation: in section 2, $\mathbf{Q}^{(i)}$ was introduced as a matrix containing the *inner products* of neighbourhood vectors, centered around the sample $\mathbf{x}_i$ under consideration. This $k \times k$ matrix has rank $\min(d, k)$ and can be written as $\mathbf{Q}^{(i)} = (\mathbf{N} - \mathbf{x}_i\mathbf{1})^T(\mathbf{N} - \mathbf{x}_i\mathbf{1})$, where $\mathbf{N}$ contains the neighbours of $\mathbf{x}_i$ as its columns and $\mathbf{1}$ is a $1 \times k$ vector containing ones. The related $d \times d$ matrix of *outer products*, $\mathbf{C}^{(i)} = (\mathbf{N} - \mathbf{x}_i\mathbf{1})(\mathbf{N} - \mathbf{x}_i\mathbf{1})^T$, is similar to a covariance matrix, but with $\mathbf{x}_i$ as mean rather than the neighbourhood mean.

It can be shown that the non-zero eigenvalues of $\mathbf{Q}^{(i)}$ and $\mathbf{C}^{(i)}$ are identical (see Appendix), although the eigenvectors are not. Intuitively, the difference between the eigenvectors of $\mathbf{Q}^{(i)}$ and $\mathbf{C}^{(i)}$ is that those of $\mathbf{Q}^{(i)}$ show how the axes can best be reconstructed using a linear combination of the samples, whereas those of $\mathbf{C}^{(i)}$ show how the samples can best be reconstructed using a linear combination of axes.

### 3.1 Intrinsic dimensionality

In traditional principal component analysis (PCA, [10]), a subspace is found by performing an eigenanalysis on the sample covariance matrix $\mathbf{S}$ and retaining the eigenvectors $\mathbf{e}_i$ corresponding to the $m$ largest eigenvalues $\lambda_i$. Rather than specifying $m$, in practice one often specifies an amount $v$ of variance that should be retained by projecting the data set: $v \leq (\sum_{j=1}^{m} \lambda_j)/(\sum_{j=1}^{d} \lambda_j)$. For a specified $v$, one can calculate a minimal $m$. Lacking knowledge of the intrinsic dimensionality, this is a more justified and intuitive method of selecting $m$, as $v$ expresses our estimate of the signal-to-noise ratio of the data. Usually, $v$ is set to 0.90 or 0.95.

In LLE, if we perform an eigenanalysis on every local covariance matrix $\mathbf{Q}^{(i)}$, the same non-zero eigenvalues will be obtained as those found by eigenanalysis on $\mathbf{C}^{(i)}$. In this way, for each sample $\mathbf{x}_i$ an $m_i$ can be found by specifying $v$. To obtain an overall $m$, one can use majority voting over the entire data set. The added computational cost is relatively small, compared to the cost of the eigenanalysis in stage II. Furthermore, it has some additional uses, which will be discussed next.

### 3.2 Automatic regularisation

Section 2 already discussed how the regularisation parameter $r$ plays an important role. Optimal values can vary over a wide range: for the mapping shown in figure 1(b), $r$ was set to $3 \times 10^{-3}$, but for an experiment mapping images of faces ($d = 100$, after global PCA) the optimal value for $r$ is in the order of $10^5$. This large difference can only partly be explained by differences in scale.

An estimate of the regularisation parameter can be found by again noting the link between $\mathbf{Q}^{(i)}$ and $\mathbf{C}^{(i)}$. In probabilistic PCA [17], the subspace $(\mathbf{e}_1, \ldots, \mathbf{e}_m)$ is estimated using traditional PCA eigenanalysis. To obtain a fully probabilistic model, i.i.d. Gaussian noise is assumed outside the subspace, of which the $\sigma^2$ is estimated by

$\hat{\sigma}^2 = \frac{1}{d-m}\sum_{j=m+1}^{d}\lambda_j$, i.e. the average eigenvalues of eigenvectors not used in constructing the subspace.

In LLE a PCA of the neighbourhoods is not explicitly calculated, but global optimisation is performed over all neighbourhoods. Still, the goal is to embed each of these neighbourhoods linearly in an $m$-dimensional space, and discard noise outside this space. Following probabilistic PCA, the matrix $\mathbf{Q}^{(i)}$ can therefore be regularised as $\mathbf{Q}^{(i)} + \hat{\sigma}_i^2\mathbf{I}$. That is, the regularisation parameter is estimated for each neighbourhood independently.

Figure 2(a) shows the data in figure 1(a) mapped using automatic regularisation and intrinsic dimensionality detection (using $v = 0.90$). The mapping looks slightly better than that of original LLE (figure 1(b)). The $\hat{\sigma}_i^2$'s found are plotted in figure 2(b), as a function of the $x$-axis of figure 1(a).

### 3.3 Subset selection

The main computational cost of LLE is incurred in the eigenanalysis of $\mathrm{M}$, which is an $n \times n$ matrix (albeit a sparse one). To lower this cost, the number of samples $n$ could be lowered. For that purpose, a method of selecting samples to be discarded is needed.

Above, the local noise estimates $\hat{\sigma}_i^2$ were used to regularise the local matrices $\mathbf{Q}^{(i)}$. These estimates can also be used to intelligently select samples to discard. If $\hat{\sigma}_i^2$ is low, $\mathbf{x}_i$ can be reconstructed fairly accurately using its neighbours, so it adds little information. If $\hat{\sigma}_i^2$ is high on the other hand, $\mathbf{x}_i$ is likely to lie either in a curved area on the manifold or not on a manifold at all, and will be dangerous to discard. A suggested process for discarding samples therefore is:

1. calculate $\hat{\sigma}_i^2, \forall \mathbf{x}_i$ and normalise: $s_i = \hat{\sigma}_i^2 / \sum_{i=1}^{n} \hat{\sigma}_i^2$;
2. use $s$ to create an empirical cumulative distribution function and *keep* $p$ samples by drawing from it.

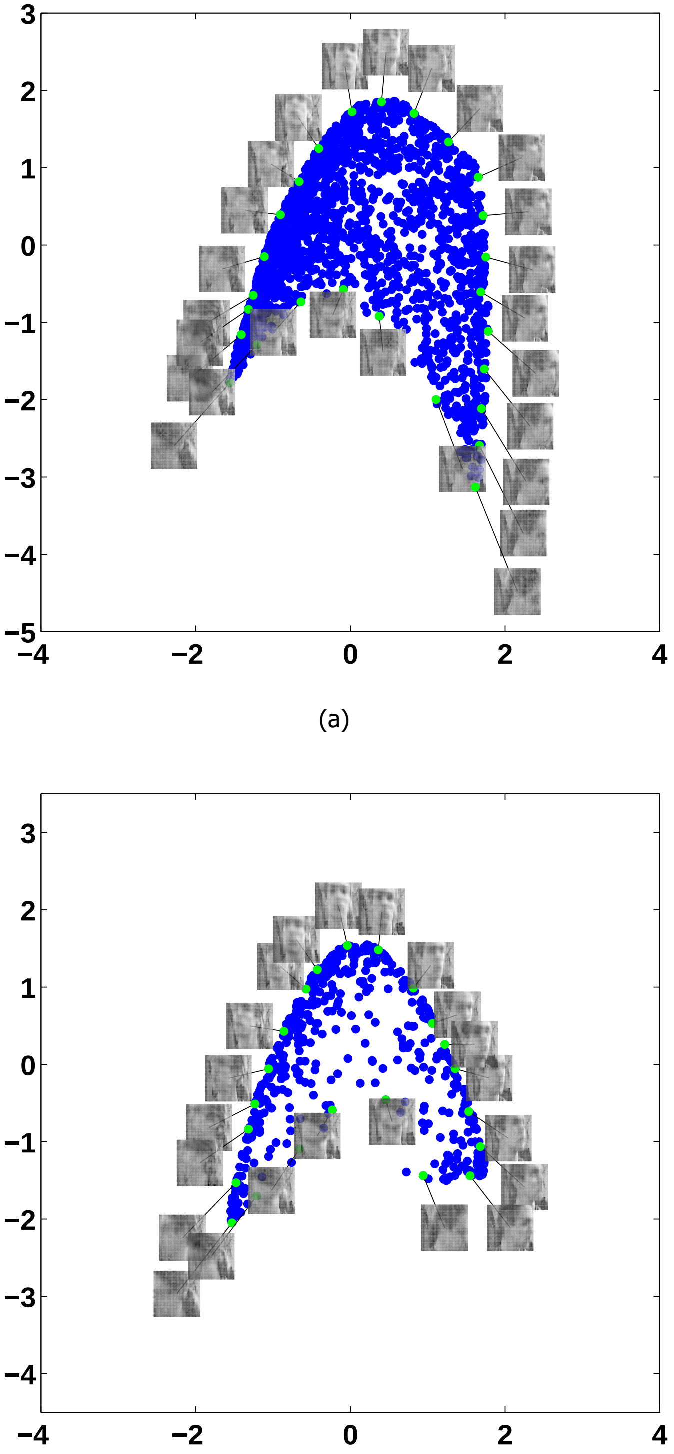
This will favour keeping samples with high $\hat{\sigma}_i^2$, but still sample the entire data set. Only when $\hat{\sigma}_i^2 = 0$ (which is unlikely for real data) will $\mathbf{x}_i$ always be discarded.

A disadvantage is that stage I of the algorithm will have to be run twice, once using all samples and once again using the selected samples (as the neighbourhood relations change). However, as the eigenanalysis in stage II is the computationally heaviest problem, the overall algorithm is likely to become faster. Furthermore, mapping new points (which has complexity $\mathcal{O}(nd)$) becomes faster as well.

Figure 2(c) shows a subset of $p = 300$ samples selected using the method outlined above. Note how more samples have been retained in areas on the manifold where there is large curvature, corresponding to the peaks in figure 2(b). The mapped data is shown in figure 2(d). Although some distortion is introduced on the right hand side, the mapping basically stays the same.

### 3.4 Summary

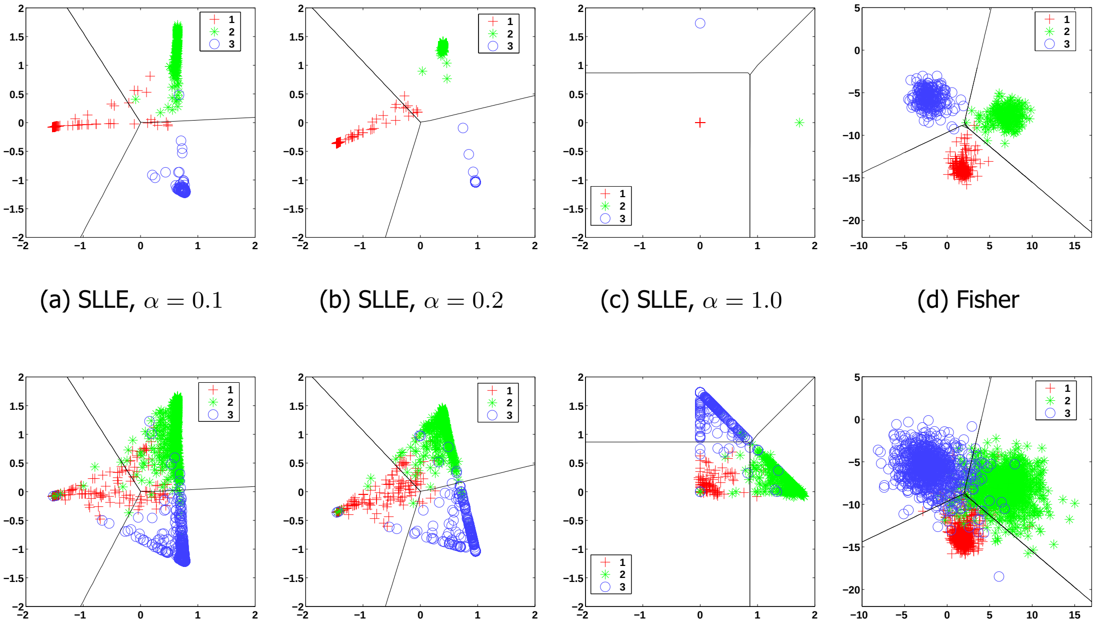
The changes proposed above lead to an algorithm with two parameters: $k$, controlling the nonlinearity of the mapping; and $v$, controlling the number of dimensions. No further reduction in the number of parameters is possible without making stronger assumptions



(a)

(b)

**Fig. 3.** (a) LLE-mapped $\mathbf{Y}$ ($n = 2000$, $m = 2$, $k = 25$, $r$ found automatically) of a face image data set. Some samples on the border of the mapping are shown as images. (b) Same, for a selected subset ($p = 500$).



(a) SLLE, $\alpha = 0.1$

(b) SLLE, $\alpha = 0.2$

(c) SLLE, $\alpha = 1.0$

(d) Fisher

(e) $e = 0.026$

(f) $e = 0.017$

(g) $e = 0.011$

(h) $e = 0.058$

**Fig. 4.** (a)-(c) NIST training set ($d = 256$, $n = 750$, $c = 3$) mapped to $m = 2$ dimensions using SLLE, for various values of $\alpha$. A trained nearest mean classifier is drawn in the figures. Note how many samples are mapped onto the same points, especially in (c). (d) Fisher mapping of the same data set, with trained nearest mean classifier. (e)-(h) Same, showing the mapped test data and test errors $e$ for the classifier shown.

on the data distribution. A third, optional parameter is $p$, the number of samples to retain in the mapping.

In principle, LLE can be applied as any linear mapping method. Figure 3(a) shows a mapping of a data set containing $n = 2000$, $40 \times 40$ pixel face images as used in [18]. The data has been mapped down to $d = 100$ dimensions using global PCA first, and contains 2 degrees of freedom: left-right and up-down viewing. For visualisation, $m$ was set to 2 (corresponding to an average $v$ of 0.60; with $v$ set to $0.90$, the method found $m = 7$ to be optimal). Regularisation was automatic, and $k$ was set to $25$. In figure 3(b), a subset of 500 samples was selected and mapped. Although the mapped data does not uniformly fill the 2D space, both mappings clearly exhibit the two degrees of freedom.

## 4 LLE for classification

In this section, the applicability of LLE in classification experiments will be discussed. The main data set used in experiments is a pre-processed version of the NIST handwritten digit data set, containing $16 \times 16$ pixel images of digits "0" to "9" [19,4]. From this set, up to 500 samples per class were selected as a training set, and 1000 samples per class were used for testing. All experimental results shown are averages and standard deviations of 10 repetitions.

### 4.1 Feature extraction

In a first experiment, LLE was compared to PCA as a feature extractor. The training set consisted of 500 samples per class. For LLE, $k = 30$ and automatic regularisation gave good results. After mapping to $m$ dimensions, a number of classifiers was trained on the mapped training data and tested: a linear discriminant, a quadratic discriminant and a 1-nearest neighbour classifier. Figures 5(a)-(c) shows performances on the test set as a function of $m$.

It is clear that LLE is useful for small numbers of dimensions; for $m$ larger than 6 to 9, the classifiers perform better on PCA-mapped data. This is because PCA, for small $m$, will introduce overlap in the mapping where there is none in the original data. As the number of dimensions increases, PCA will discard less and less information. At the same time, LLE will start overfitting, a problem to which it is much more sensitive than PCA because of its nonlinear nature.

Given these observations, LLE might not seem to be very useful in classification. It performs better than PCA when mapping down to a very small number of dimensions, at which performance of any classifier is relatively poor. It might still be useful for visualisation. However, its higher computational cost of mapping new samples ($\mathcal{O}(nd)$ vs. $\mathcal{O}(d)$ for PCA) does not seem to warrant application of standard LLE as part of a classification solution.

### 4.2 Supervised LLE

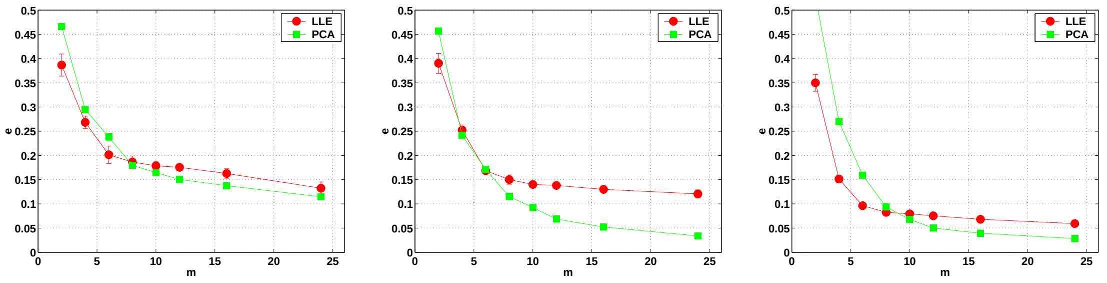
Until now, LLE has been used as an unsupervised technique. However, the LLE problem can easily be rephrased to use class label information, $\omega_i \in \Omega$ ($|\Omega| = c$) with each $\mathbf{x}_i$, during training. The idea is to find a mapping separating within-class structure from between-class structure. The easiest way to do this is to select the neighbours $\mathbf{x}_{N(.)}$ in eqn. 2 from just the class that $\mathbf{x}_i$ itself belongs to.

A slightly more complicated method would be by using the distance matrix formulation as in eqn. 5, but adding distance between samples in different classes:

$$\mathbf{D}' = \mathbf{D} + \alpha \max(\mathbf{D}) \mathbf{\Delta} \tag{8}$$

where $\Delta_{jm} = 1$ if $\omega_j \neq \omega_m$, and $0$ otherwise. In this formulation, $\alpha \in (0, 1)$ controls the amount to which class information should be incorporated. A data set is created in which there are $c$ "disconnected" classes, each of which should be mapped fairly by LLE. These added degrees of freedom are then used by stage II to separate the classes, using eigenvectors 2 to $c$, just as the first eigenvector was used to discard the mean of the data. Mapped classes are separated due to the constraint $\frac{1}{n}\mathbf{Y}\mathbf{Y}^T = \mathbf{I}$. After the first $c$ eigenvectors, the next $c \cdot m$ eigenvectors correspond to LLE mappings optimal for the various individual classes.

In effect, this supervised version of LLE behaves as a nonlinear Fisher mapping, where $\alpha$ controls the nonlinearity. It will be denoted by SLLE. Figure 4 illustrates this on digits "1", "2" and "3" of the NIST data set, comparing SLLE-mapped data to Fisher-mapped data. For $\alpha = 1.0$ (figure 4(c)), SLLE succeeds in mapping all samples in a class onto a single point. All new samples (figure 4(g)) will be linearly interpolated between these class points. Performance of the simple nearest mean classifier trained on the



(a) (b) (c)

**Fig. 5.** Test error of the (a) linear discriminant, (b) quadratic discriminant and (c) 1-nearest neighbour classifier trained on LLE-mapped vs. PCA-mapped data, as a function of $m$ ($n = 5000$, $c = 10$, $d = 256$, $k = 30$, automatic regularisation).

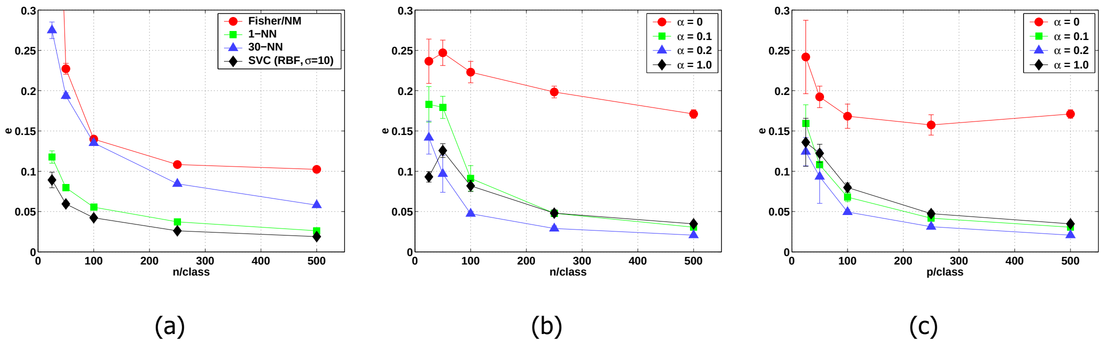
mapped data is remarkably good. As a comparison, the support vector classifier (RBF kernel, $\sigma = 10$) classifier on the original data ($d = 256$) reaches a test error of $e = 0.011$, the 1-nearest neighbour classifier gives $e = 0.010$.

SLLE was also applied to the full NIST data set, and performance of the nearest mean classifier trained on the mapped data was compared to that of a number of classifiers trained on the original data. The parameter $v$ was set to 0.90, giving values of $m$ in the range 9-12. Figure 6 shows the results. SLLE's performance is slightly worse than that of the best classifier on this data set, the SVC [4], but slightly better than that of the 1-nearest neighbour classifier. For $n$ = 500/class and $\alpha = 0.2$, SLLE followed by a nearest mean classifier reaches a test error $e = 0.023$, whereas for the best-performing SVC on the original data this number is $e = 0.019$ and for the 1-nearest neighbour classifier $e = 0.027$.

Choosing $\alpha$ such that class label information is not used fully (e.g. in the range 0.1-0.2) yields the best generalisation by SLLE. This indicates that the combination of SLLE (based on nearest neighbours) and the nearest mean classifier can effectively control the nonlinearity of ordinary nearest-neighbour based classifiers, at a comparable computational cost. An added advantage of SLLE is that overlap between each pair of classes can easily be visualised, as classes are mapped on the vertices of a $(c-1)$-dimensional hypercube (cf. figures 4(e)-(g)).

For the $k$-nearest neighbour classifier (figure 6(a)), $k = 1$ was selected by cross-validation. To illustrate that choosing a moderate $\alpha$ for SLLE does not simply correspond to choosing a different $k$, the – significantly worse – performance of the 30-nearest neighbour classifier is also given in figure 6(a).

Finally, the subset selection method discussed in the previous section was also applied to SSLE; results are shown in figure 6(c). It only yields better results than random selection (figure 6(b)). for small subsets, $p$ = 25-50/class. Apparently, SLLE is not very sensitive to the exact choice of samples for reasonably large training sets.



(a) (b) (c)

**Fig. 6.** (a) Test error for a number of classifiers trained on the original data: Fisher mapping followed by a nearest mean classifier, 1-nearest neighbour, 30-nearest neighbour and an RBF support vector classifier ($\sigma = 10$). (b) Test error of the nearest mean classifier trained on SLLE-mapped data for various $\alpha$, as a function of $n$ ($v$ = 0.90, $k$ = 30, automatic regularisation). (c) Same, as a function of selected subset size $p$ for $n$ = 500.

## 5 Conclusions

An extension of LLE to supervised mapping, SLLE, was discussed. SLLE introduces an additional parameter $\alpha$, controlling how much of the class label information should be taken into account. Experiments showed that training a simple classifier on the mapped data gave results comparable to optimised state-of-the-art classifiers on a handwritten digit data set.

A number of enhancements to locally linear embedding were also proposed: specifying the amount of variance to retain, $v$, to find the intrinsic dimensionality of the data; automatic setting of the optimisation parameter $r$; and subset selection by discarding samples in areas of low curvature. These enhancements were shown to perform well on a set of face images and to be useful in the extension to supervised mapping.

The method still has a number of parameters which need to be set, among which $k$ is the most important. However, LLE will always need at least one parameter controlling its nonlinearity. Either $m$ is specified and $k$ is found automatically [13], or $k$ is fixed and $m$ is optimised, as in this paper. An interesting topic for further research is how to bring the computational complexity of mapping new samples down further, using a method such as $k$-AESA [14]. It would also be interesting to compare the proposed method to other combinations of nonlinear mapping methods and classifiers on a number of data sets, to gain insight into what methods are suitable for what class of problems.

## Acknowledgements

The authors would like to thank Sjaak Verbeek for fruitful discussion and providing us the face image data set. Thanks also go to Oleg Okun for a useful exchange of ideas. For training support vector classifiers we used the Torch package [3]. This work was partly sponsored by the Foundation for Applied Sciences (STW) under project number AIF.4997.

## Appendix

Given a neighbourhood with samples stored as columns in a matrix $\mathbf{N}$, eigenanalysis on $\mathbf{Q} = \mathbf{N}^T\mathbf{N}$ and $\mathbf{C} = \mathbf{N}\mathbf{N}^T$ will result in the same eigenvalues. For $\mathbf{Q}$, the eigenvector equation is:

$$\mathbf{Q}\mathbf{e}_i = \mathbf{N}^T\mathbf{N}\mathbf{e}_i = \lambda_i\mathbf{e}_i \tag{9}$$

and for $\mathbf{C}$,

$$\mathbf{C}\mathbf{f}_i = \mathbf{N}\mathbf{N}^T\mathbf{f}_i = \gamma_i\mathbf{f}_i. \tag{10}$$

Pre-multiplying eqn. 9 by $\mathbf{N}$ gives

$$\mathbf{N}\mathbf{N}^T\mathbf{N}\mathbf{e}_i = \lambda_i\mathbf{N}\mathbf{e}_i, \tag{11}$$

which shows that eqns. 9 and 10 are equal when $\mathbf{f}_i = \mathbf{N}\mathbf{e}_i$ and $\lambda_i = \gamma_i$.

## References

1. C.M. Bishop, M. Svensén, and C.K.I. Williams. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998.
2. I. Borg and P. Groenen. *Modern multidimensional scaling*. Springer-Verlag, Berlin, 1997.
3. R. Collobert and S. Bengio. SVMTorch: support vector machines for large-scale regression problems. *Journal of Machine Learning Research*, 1:143–160, 2001.
4. D. de Ridder. *Adaptive methods of image processing*. PhD thesis, Delft University of Technology, Delft, 2001.
5. D. DeMers and G.W. Cottrell. Non-linear dimensionality reduction. In C.L. Giles, S.J. Hanson, and J.D. Cowan, editors, *Advances in Neural Information Processing Systems 5*, pages 580–587. Morgan Kaufmann, San Mateo, CA, 1993.
6. P.A. Devijver and J. Kittler. *Pattern recognition, a statistical approach*. Prentice-Hall, London, 1982.
7. R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern classification*. John Wiley & Sons, New York, NY, $2^{nd}$ edition, 2001.
8. K. Fan. On a theorem of Weyl concerning eigenvalues of linear transformations. In *Proc. of the National Academy of Sciences*, volume 35, pages 652–683, Washington, DC, 1949. PNAS.
9. T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84:502–516, 1989.
10. H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.
11. A.K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.
12. T. Kohonen. *Self-organizing maps*. Springer Series in Information Sciences. Springer-Verlag, Berlin, 1995.
13. O. Kouropteva, O. Okun, and M. Pietikäinen. Selection of the optimal parameter value for the locally linear embedding algorithm. 2002. Submitted to $1^{st}$ International Conference on Fuzzy Systems and Knowledge Discovery.

14. L. Mico, J. Oncina, and E. Vidal. A new version of the nearest-neighbour approximating and eliminating search algorithm (AESA) with linear preprocessing time and memory requirements. *Pattern Recognition Letters*, 15(1):9–17, 1994.
15. L.K. Saul and S.T. Roweis. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.
16. M.E. Tipping and C.M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999.
17. M.E. Tipping and C.M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61(3):611–622, 1999.
18. J.J. Verbeek, N. Vlassis, and B. Kröse. Coordinating Principal Component analyzers. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pages 914–919, Madrid, Spain, 2002.
19. C. L. Wilson and M. D. Garris. Handprinted character database 3, february 1992. National Institute of Standards and Technology; Advanced Systems division.