Pattern Recognition Letters 32 (2011) 1029–1035

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Guided Locally Linear Embedding

Babak Alipanahi [a], Ali Ghodsi [b,*]

[a] David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada N2L 3G1
[b] Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, ON, Canada N2L 3G1

ARTICLE INFO

Article history:
Received 26 August 2010
Available online 21 March 2011
Communicated by M.A. Girolami

Keywords:
Supervised dimensionality reduction
Locally Linear Embedding
Classification
Pattern recognition

ABSTRACT

Nonlinear dimensionality reduction is the problem of retrieving a low-dimensional representation of a manifold that is embedded in a high-dimensional observation space. Locally Linear Embedding (LLE), a prominent dimensionality reduction technique is an unsupervised algorithm; as such, it is not possible to guide it toward modes of variability that may be of particular interest. This paper proposes a supervised variation of LLE. Similar to LLE, it retrieves a low-dimensional global coordinate system that faithfully represents the embedded manifold. Unlike LLE, however, it produces an embedding in which predefined modes of variation are preserved. This can improve several supervised learning tasks including pattern recognition, regression, and data visualization.

1. Introduction

Conventional classification and pattern recognition methods fail to produce satisfactory results when applied to high-dimensional data sets. This is partially due to the problem of the "curse of dimensionality". One approach to this problem is to reduce the dimensionality of the data by projecting it onto a subspace or submanifold. There exist many different methods to retrieve a low-dimensional global coordinate set that faithfully represents the embedded manifold of the high-dimensional observation space. One of the limitations of most prominent dimensionality reduction techniques, however, is that they are unsupervised. Therefore, it is not possible to guide them towards modes of variability that may be of particular interest. For example, consider a set of several images of men and women, with and without glasses. One mode of variation in this data set is gender; another is the presence or absence of glasses. However, there is no way to guide Locally Linear Embedding (LLE) (Roweis and Saul, 2000; Saul and Roweis, 2003) toward one of these modes of variation. LLE captures the dominant mode of variation, which could be neither of these two characteristics. This problem can be mitigated by the proper exploitation of "side-information" about the data set. For example, one could apply labels to a subset of the data to indicate the type of variability that is of interest. Then the algorithm could be encouraged to reflect this kind of variability.

In some existing methods, side-information about the manifold can be provided in the very restrictive form of equivalence constraints which indicate data points that belong together in the low-dimensional embedding. Alternatively, the proposed approach can employ a much wider range of constraints and side-information including one-dimensional or even multi-dimensional qualitative and quantitative target variables. This extends the use of this method to the large class of regression and function estimation problems.

There have been several attempts to develop a supervised version of LLE (de Ridder et al., 2003; Kouropteva et al., 2003; Li et al., 2008; Zhang, 2009; Zhang and Zhao, 2007; Zhao et al., 2005). Most of these methods rely on class labels for the data, in order to modify the within and between class distances. In α-Supervised LLE (α-SLLE) (de Ridder et al., 2003; Kouropteva et al., 2003), the between-class distances are increased by a constant value, but within-class distances remain unchanged. Enhanced SLLE (Zhang, 2009) applies a nonlinear function to pairwise distances which limits within-class distances to 1, but increases the between-class distances. The method proposed in Zhao et al. (2005) set the between-class distances to infinity. As a result, each point chooses all neighbors from the same class.

We propose a different approach, inspired by the newly introduced independence measure called the Hilbert–Schmidt Independence Criterion (HSIC) (Gretton et al., 2005, 2008). We will show that conventional LLE can be viewed as a dependence maximization method. That is, we show that LLE will find a low-dimensional representation of the data that has a maximum dependence to the high-dimensional observation space. We then modify the LLE objective function in order to find a low-dimensional representation that simultaneously depends on both the observed data as well as a set of target variables.

The rest of this paper is organized as follows: in Section 2, LLE is briefly reviewed and its connection with Kernel PCA is described.

* Corresponding author. Fax: +1 519 746 1875.
E-mail address: aghodsib@math.uwaterloo.ca (A. Ghodsi).

0167-8655/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2011.02.002

The Hilbert–Schmidt Independence Criterion is described in Section 3. The proposed method, which we call Guided LLE (GLLE), is presented in Section 4. Section 5 illustrates the uses of the proposed algorithm when applied to real data sets. Finally, we conclude in Section 6.

1.1. Notation

We use $\mathbb{R}^p$ and $\mathscr{M}^{p\times q}$ to denote the space of real p-dimensional vectors and real $p \times q$ matrices, respectively. Scalars, vectors, and matrices are shown in small, small bold, and capital letters, respectively. We use $\|\boldsymbol{x}\|$ to denote the Euclidean norm (ℓ_2-norm) of vector $\boldsymbol{x}$. For a square matrix $A \in \mathscr{M}^{p\times p}$, $\boldsymbol{Tr}(A)$ is the sum of its diagonal elements. For a matrix A, A_{ij} denotes its (i,j)th entry. I_p, $\boldsymbol{1}_p$, and $\boldsymbol{0}_p$, denote identity matrix, all-one, and all-zero vector, respectively. For any matrix A, $A^\top$ and $A^\dagger$ denote its transpose and pseudo-inverse, respectively.

2. Locally Linear Embedding

Locally Linear Embedding (LLE) computes a low-dimensional neighborhood-preserving embedding of a high-dimensional dataset (Roweis and Saul, 2000). A dataset of dimensionality p, which is assumed to lie on or near a smooth nonlinear manifold of dimensionality $d \ll p$, is mapped onto a single global coordinate system of lower dimensionality, d. The global nonlinear structure is recovered by locally linear fits.

Consider n real-valued vectors $\{\boldsymbol{x}_i \in \mathbb{R}^p\}_{i=1}^n$ sampled from some underlying manifold. We can assume each data point and its neighbors lie on, or are close to, a locally linear patch of the manifold. By a linear mapping, consisting of a translation, rotation, and scaling, the high-dimensional coordinates of each neighborhood can be mapped to the global internal coordinates along the manifold. Therefore, the nonlinear structure of the data can be identified through two linear steps: first, by computing the locally linear patches, and secondly, by computing the linear mapping to the coordinate system of the manifold.

The main goal here is to map the high-dimensional data points to the single global coordinate system of the manifold such that the relationships between neighboring points are preserved. This proceeds in three steps:

1. Identify the neighbors of each data point $\boldsymbol{x}_i$. This can be done by finding the k nearest neighbors, or by choosing all points within some fixed radius, ϵ.
2. Compute the weights that best linearly reconstruct $\boldsymbol{x}_i$ from its neighbors.
3. Find the low-dimensional embedding vector $\{\boldsymbol{y}_i \in \mathbb{R}^d\}_{i=1}^n$, which is best reconstructed by the weights determined in the previous step.

After finding the nearest neighbors in the first step, the second step must compute a local geometry for each locally linear patch. This geometry is characterized by linear coefficients that reconstruct each data point from its neighbors:

$$\min_W \sum_{i=1}^n \left\| \boldsymbol{x}_i - \sum_{j=1}^k W_{i\eta_i(j)} \boldsymbol{x}_{\eta_i(j)} \right\|^2, \tag{1}$$

where $\eta_i(j)$ is the index of the jth neighbor of $\boldsymbol{x}_i$, and $W \in \mathscr{M}^{n\times n}$ is the weight matrix. It then selects code vectors so as to preserve the reconstruction weights by solving:

$$\min_Y \sum_{i=1}^n \left\| \boldsymbol{y}_i - \sum_{j=1}^k W_{i\eta_i(j)} \boldsymbol{y}_{\eta_i(j)} \right\|^2. \tag{2}$$

This objective can be reformulated as:

$$\min_Y \boldsymbol{Tr}(YMY^\top), \tag{3}$$

where $Y = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n]$ and $M = (I - W)^\top (I - W)$.

The solution for Y can have an arbitrary origin and orientation. In order to make the problem well-posed, these two degrees of freedom must be removed. Requiring the coordinates to be centered on the origin ($\sum_i \boldsymbol{y}_i = \boldsymbol{0}$), and constraining the embedding data points to have unit covariance ($YY^\top = I$), removes the first and second degrees of freedom, respectively.

The cost function can be optimized initially by the second of these two constraints. Under this constraint, the cost is minimized when the columns of $Y^\top$ (rows of Y) are the eigenvectors associated with the lowest eigenvalues of M. Discarding the eigenvector associated with eigenvalue 0 satisfies the first constraint.

2.1. Connection with Kernel PCA

A straightforward connection between LLE and Kernel PCA has been shown in (Schölkopf and Smola, 2002; Bengio and Paiement, 2003). Let $\lambda_{\max}$ be the largest eigenvalue of $M = (I - W)^\top (I - W)$. Then define the LLE kernel to be:

$$K := (\lambda I - M). \tag{4}$$

This kernel is, in fact, a similarity measure based on the similarity of the weights required to reconstruct two patterns in terms of k neighboring patterns. The leading eigenvector of K is $\boldsymbol{1}_n$, and the eigenvectors $2, \ldots, d+1$ provide the LLE embedding.

An alternative interpretation of LLE as a specific form of Kernel PCA has been discussed in (Ham et al., 2004) in detail. Based on this discussion, performing Kernel PCA on the pseudo-inverse of M is equivalent to LLE. Therefore, the LLE kernel can also be defined as:

$$K_\dagger := M^\dagger. \tag{5}$$

3. Hilbert–Schmidt Independence Criterion

The Hilbert–Schmidt norm of the cross-covariance operator (Gretton et al., 2005, 2008) in reproducing kernel Hilbert spaces (RKHS) (Schölkopf and Smola, 2002) has been proposed as an independence criterion. This measure will be referred to as the Hilbert–Schmidt Independence Criterion, or HSIC. HSIC uses the fact that two random variables $\boldsymbol{x}$ and $\boldsymbol{y}$ are independent if and only if any bounded continuous function of the two random variables is uncorrelated. Consider two multivariate random variables $\boldsymbol{x}$ and $\boldsymbol{y}$. Define a RKHS $\mathscr{F}$ from $\mathscr{X}$ to $\mathbb{R}$ containing all continuous bounded real-valued functions of $\boldsymbol{x}$, and a RKHS $\mathscr{G}$ from $\mathscr{Y}$ to $\mathbb{R}$ containing all continuous bounded real-valued functions of $\boldsymbol{y}$. Here, $\mathscr{X}$ and $\mathscr{Y}$ denote the support (the set of possible values) of the random variables $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. We are interested in the cross-covariance between elements of $\mathscr{F}$ and $\mathscr{G}$:

$$\mathrm{Cov}(f(\boldsymbol{x}), g(\boldsymbol{y})) = \boldsymbol{E}_{\boldsymbol{xy}}[f(\boldsymbol{x})g(\boldsymbol{y})] - \boldsymbol{E}_{\boldsymbol{x}}[f(\boldsymbol{x})]\boldsymbol{E}_{\boldsymbol{y}}[g(\boldsymbol{y})]. \tag{6}$$

It can be shown that there exists a unique operator[1] $C_{\boldsymbol{xy}} : \mathscr{G} \to \mathscr{F}$, mapping elements of $\mathscr{G}$ to elements of $\mathscr{F}$ such that: $\langle f, C_{\boldsymbol{xy}}(g)\rangle_{\mathscr{F}} = \mathrm{Cov}(f, g)$ for all $f \in \mathscr{F}$ and $g \in \mathscr{G}$. This operator is called the cross-covariance operator.

[1] In the terminology of functional analysis, an operator is a mapping which maps elements from one Hilbert space to elements of another Hilbert space.

The measure of dependence between two random variables can be defined as the Hilbert–Schmidt norm[2] of the cross-covariance operator:

$$\mathrm{HSIC}(p_{\boldsymbol{xy}}, \mathscr{F}, \mathscr{G}) := \|C_{\boldsymbol{xy}}\|^2_{\mathrm{HS}}. \tag{7}$$

Note that if $\|C_{xy}\|^2_{\mathrm{HS}}$ is zero, then the value of $\langle f, C_{\boldsymbol{xy}}(g)\rangle$, i.e., $\mathrm{Cov}(f,g)$, will always be zero for any $f \in \mathscr{F}$ and $g \in \mathscr{G}$, and thus the random variables $\boldsymbol{x}$ and $\boldsymbol{y}$ are independent.

3.1. Empirical HSIC

To compute the HSIC we need to express it in terms of kernel functions. This can be achieved via the following identity:

$$\begin{aligned}\mathrm{HSIC}(p_{\boldsymbol{xy}}, \mathscr{F}, \mathscr{G}) = {} & \boldsymbol{E}_{\boldsymbol{x},\boldsymbol{x}',\boldsymbol{y},\boldsymbol{y}'}[k_{\boldsymbol{x}}(\boldsymbol{x},\boldsymbol{x}')k_{\boldsymbol{y}}(\boldsymbol{y},\boldsymbol{y}')] \\ & + \boldsymbol{E}_{\boldsymbol{x},\boldsymbol{x}'}[k_{\boldsymbol{x}}(\boldsymbol{x},\boldsymbol{x}')]\boldsymbol{E}_{\boldsymbol{y},\boldsymbol{y}'}[k_{\boldsymbol{y}}(\boldsymbol{y},\boldsymbol{y}')] \\ & - 2\boldsymbol{E}_{\boldsymbol{x},\boldsymbol{y}}[\boldsymbol{E}_{\boldsymbol{x}'}[k_{\boldsymbol{x}}(\boldsymbol{x},\boldsymbol{x}')]\boldsymbol{E}_{\boldsymbol{y}'}[k_{\boldsymbol{y}}(\boldsymbol{y},\boldsymbol{y}')]].\end{aligned} \tag{8}$$

Now let $\mathscr{Z} := \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n \subseteq \mathscr{X} \times \mathscr{Y}$ be a series of n independent observations drawn from $p_{\boldsymbol{xy}}$. An estimator of HSIC is given by:

$$\mathrm{HSIC}(\mathscr{Z}, \mathscr{F}, \mathscr{G}) := (n-1)^{-2}\,\boldsymbol{Tr}(K_{\boldsymbol{x}}HK_{\boldsymbol{y}}H), \tag{9}$$

where H, $K_{\boldsymbol{x}}$, and $K_{\boldsymbol{y}}$ are $n \times n$ positive semidefinite matrices, $(K_{\boldsymbol{x}})_{ij} := k_{\boldsymbol{x}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $(K_{\boldsymbol{y}})_{ij} := k_{\boldsymbol{y}}(\boldsymbol{y}_i, \boldsymbol{y}_j)$, $k_{\boldsymbol{x}}(\cdot,\cdot)$ and $k_{\boldsymbol{y}}(\cdot,\cdot)$ are positive semi definite kernel functions (see Schölkopf and Smola (2002) for details on kernel functions), and $H = \left(I - \frac{1}{n}\boldsymbol{1}\boldsymbol{1}^\top\right)\left(I - \frac{1}{n}\boldsymbol{1}\boldsymbol{1}^\top\right)$ is the centering matrix. Based on this result, in order to maximize the dependence between two random variables $\boldsymbol{x}$ and $\boldsymbol{y}$, we need to increase the value of the empirical estimate, i.e., $\boldsymbol{Tr}(K_{\boldsymbol{x}}HK_{\boldsymbol{y}}H)$.

4. Guided LLE

Many unsupervised dimensionality reduction algorithms including LLE can be interpreted as dependence maximization. Here, kernel $K_{\boldsymbol{y}}$ in (9) denotes the kernel of the low-dimensional representation of the data, while $K_{\boldsymbol{x}}$ represents the kernel of the data in high-dimension. More precisely, if we assume the original data points $\boldsymbol{x}_i$ and embedded data points $\boldsymbol{y}_i$ are multi-variate random variables, and $\mathscr{Z} := \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n \subseteq \mathscr{X} \times \mathscr{Y}$ are independently sampled from $p_{\boldsymbol{xy}}$, then the empirical HSIC between $\boldsymbol{x}$ and $\boldsymbol{y}$ is calculated as:

$$\mathrm{HSIC}(\mathscr{Z}, \mathscr{F}, \mathscr{G}) = \boldsymbol{Tr}(K_{\boldsymbol{y}}HK_{\boldsymbol{x}}H). \tag{10}$$

In the case that $K_{\boldsymbol{x}} = K_\dagger$, the LLE kernel, and $K_{\boldsymbol{y}} = Y^\top Y$ is the linear kernel, then maximizing (10) subject to the constraint that $YY^\top = I$ resembles the formulation of LLE in (3):

$$\boldsymbol{Tr}(K_{\boldsymbol{y}}HK_{\boldsymbol{x}}H) = \boldsymbol{Tr}(YHK_\dagger HY^\top). \tag{11}$$

It is important to note that the kernel $K_\dagger$ suggested for LLE is already double-centered and the application of the centering matrix H in (11) leaves it unchanged.[3] Therefore, (11) can be written as $\boldsymbol{Tr}(YK_\dagger Y^\top)$ and the optimization can be formulated as:

$$\begin{aligned}&\max_Y \ \boldsymbol{Tr}(YK_\dagger Y^\top) \\ &\text{s.t.} \quad YY^\top = I,\end{aligned} \tag{12}$$

which is equivalent to:

$$\begin{aligned}&\min_Y \ \boldsymbol{Tr}(YMY^\top) \\ &\text{s.t.} \quad YY^\top = I,\end{aligned} \tag{13}$$

since $K_\dagger$ is the pseudo-inverse of M. This means that LLE can be seen entirely as a dependence maximization problem in which the algorithm searches for a d-dimensional representation with maximum dependence to the p-dimensional ($d \ll p$) observation space.

Now consider a supervised problem (e.g. pattern recognition or function estimation), where each data point $\boldsymbol{x}_i$ is associated with a corresponding target t_i. The targets may be discrete (i.e., class labels in classification), or continuous (as in regression). If K_t is a kernel of target variables t_i, then a d-dimensional representation $\boldsymbol{y}$ with maximum dependence to t can be retrieved from the following optimization problem:

$$\begin{aligned}&\max_Y \ \boldsymbol{Tr}(YHK_tHY^\top) \\ &\text{s.t.} \quad YY^\top = I,\end{aligned} \tag{14}$$

or equivalently,

$$\begin{aligned}&\min_Y \ \boldsymbol{Tr}(YHK_t^\dagger HY^\top) \\ &\text{s.t.} \quad YY^\top = I.\end{aligned} \tag{15}$$

We will now propose a new cost function that can combine the two different d-dimensional representations in (12) and (15) into a single cost function. The first similarity measure (kernel $K_\dagger$) is derived from LLE and constructed as $((I - W_{\boldsymbol{x}})^\top(I - W_{\boldsymbol{x}}))^\dagger$, where $W_{\boldsymbol{x}}$ is the matrix of reconstruction weights of $\{\boldsymbol{x}_i\}_{i=1}^n$. A similarity measure of target values t_i, expressed in a matrix K_t, can be derived from LLE as $((I - W_t)^\top(I - W_t))^\dagger$ in the same manner where W_t is the matrix of reconstruction weights of $\{t_i\}_{i=1}^n$. This can also be derived from any arbitrary kernel function. We can now form our combined embedding objective as:

$$\begin{aligned}&\min_Y \ (1-\gamma)\boldsymbol{Tr}(Y(K_\dagger)^\dagger Y^\top) + \gamma\,\boldsymbol{Tr}(Y(K_t)^\dagger Y^\top) \\ &\text{s.t.} \quad YY^\top = I,\end{aligned} \tag{16}$$

where Y is a matrix of code vectors and $0 \leqslant \gamma \leqslant 1$. The first trace term in this objective is essentially the LLE objective, trying to preserve the structure of $\{\boldsymbol{x}_i\}_{i=1}^n$. The second trace term is again the cost function of a locality-preserving embedding; but unlike the first term, it attempts to preserve the target value information in the embedded space. The parameter γ mixes the objectives; it embeds on the basis of only $\{\boldsymbol{x}_i\}_{i=1}^n$ as γ tends to zero, and on the target value $\{t_i\}_{i=1}^n$ as γ tends to 1. This problem is an instance of the Rayleigh–Ritz theorem and can be solved in closed form. If we combine the two trace terms together and form M_ψ as

$$M_\psi = (1-\gamma)M + \gamma K_t^\dagger, \tag{17}$$

then the optimal Y, is the bottom d eigenvectors of M_ψ, with a corresponding nonzero eigenvalue. Algorithm 1 illustrates details of the Guided Locally Linear Embedding algorithm.

4.1. A short note on α-SLLE

Most of the existing variations of supervised LLE arbitrarily modify the distances between data points based on their label information. Among existing supervised LLE methods, the most cited and widely-used approach is α-SLLE (de Ridder et al., 2003). In α-SLLE, the $n \times n$ distance matrix between all points, denoted by Δ, is modified as follows:

$$\Delta' = \Delta + \alpha \max(\Delta)\Lambda, \quad \alpha \in [0,1], \tag{18}$$

where Δ' is the new distance matrix, $\max(\Delta)$ is the maximum entry in Δ, and $\Lambda_{ij} = 1$ if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ do not share the same label, and 0

[2] We may define the concept of the norm of an operator. For example, consider an operator in the form of a matrix $C_{n\times m}$ mapping vectors of $\mathbb{R}^m$ to vectors of $\mathbb{R}^n$. Then the Frobenius norm of this matrix may be defined as the norm of the corresponding operator. There are different norms defined for operators. One of them is called the Hilbert–Schmidt (HS) norm and is defined as follows: $\|C\|^2_{\mathrm{HS}} := \sum_{ij}\langle C\boldsymbol{v}_i, \boldsymbol{u}_j\rangle^2_{\mathscr{F}}$ where u_j and v_i are the orthogonal bases of $\mathscr{F}$ and $\mathscr{G}$, respectively. It is easy to see that the Frobenius norm on matrices may be considered a special case of this norm.

[3] Also, the centering effect of H on the kernel $K := (\lambda I - M)$ does not affect the result. This is because multiplication by H, from both the left and the right, only removes the eigenvector $\boldsymbol{1}$ and leaves the remaining eigenvectors unchanged.

otherwise. The 0-SLLE is identical to the original LLE. However, in 1-SLLE, neighbors of $\boldsymbol{x}_i$ are selected only from the same class. This could be problematic as it leads to a disconnected neighborhood graph; one should note that LLE needs to work with a connected graph by construction. Recall that the solution of LLE is derived from the eigendecomposition of matrix $M = (I - W)^\top (I - W)$. Here $(I - W)$ is the Laplacian of the neighborhood graph, and therefore the number of eigenvalues that are equal to zero indicates the number of connected subgraphs. For a data set with c classes, the bottom c eigenvalues of M in 1-SLLE will all be equal to zero. This will produce a low-dimensional mapping completely independent from the observation space, in which all the points within each class are collapsed to a single point. As far as a supervised task such as classification or regression is concerned, such an embedding will perform poorly at generalization. This may sound like a worst-case scenario. However, since $\max(\Delta)$ can be much larger than the local distances between close neighboring points, even for very small values of α, this algorithm does not ensure a connected graph.

Algorithm 1. Guided Locally Linear Embedding

Input: $\{(\boldsymbol{x}_i \in \mathbb{R}^p, t_i)\}_1^n,\ k,\ \gamma$
Output: $\{\boldsymbol{y}_i \in \mathbb{R}^d\}_1^n$

1: Identify the k-nearest neighbors of each data point $\boldsymbol{x}_i$.
2: Define $\eta_i(j)$ as the index of the jth neighbor of $\boldsymbol{x}_i$.
3: For each data point $\boldsymbol{x}_i$, find the $k \times k$ local Gram matrix $G^{(i)}$ such that
$$G^{(i)}_{\ell\ell'} = (\boldsymbol{x}_i - \boldsymbol{x}_{\eta_i(\ell)})^\top (\boldsymbol{x}_i - \boldsymbol{x}_{\eta_i(\ell')}).$$
4: Find the reconstruction weights $\boldsymbol{w}_i$ by solving $G^{(i)}\boldsymbol{w}_i = \boldsymbol{1}$.
5: Normalize weights by $\boldsymbol{w}_i \leftarrow \boldsymbol{w}_i / \sum \boldsymbol{w}_i$.
6: Form the sparse reconstruction weight matrix W as
$$W_{ij} = \begin{cases} \boldsymbol{w}_i(\ell) & \exists \ell :\ j = \eta_i(\ell) \\ 0 & \text{otherwise} \end{cases}$$
(This is the W that minimize (1).)
7: Construct matrix M from W as:
$$M = (I - W)^\top (I - W).$$
8: Construct a kernel matrix for targets and find its pseudo-inverse $(K_t)^\dagger$.
(This can be done for example from (22).)
9: Find Y as the d bottom eigenvectors of M_ψ with a corresponding nonzero eigenvalue where
$$M_\psi = \gamma M + (1 - \gamma) K_t^\dagger.$$
10: Set $\boldsymbol{y}_i$ as the ith row of matrix Y.

5. Experimental results

There are many different approaches in supervised dimensionality reduction algorithms. This includes classical Fisher's Discriminant Analysis (Fisher, 1936), the large family of methods known as Metric Learning (Xing et al., 2002; Bilenko et al., 2004; Chang and Yeung, 2004; Chang and Yeung, 2006; Yeung and Chang, 2006; Basu et al., 2004; Weinberger et al., 2006; Globerson and Roweis, 2006; Alipanahi et al., 2008), the family of Sufficient Dimension Reduction (SDR) (Li, 1991, 1992; Cook and Weisberg, 1991; Samarov, 1993; Cook and Yin, 2001; Hristache et al., 2001; Torkkola, 2003; Fukumizu et al., 2004), algorithms and supervised version of LLE (de Ridder et al., 2003; Kouropteva et al., 2003; Li et al., 2008; Zhang, 2009; Zhang and Zhao, 2007; Zhao et al., 2005). Due to the vast variety of these techniques, we restrict our attention to the family of supervised LLE methods and compare the proposed algorithm with α-SLLE in this family because to the best of our knowledge α-SLLE is the most cited and the most popular method among existing supervised LLE.

5.1. The kernel on targets

In order to compute M_ψ in (17), we need the pseudo-inverse of the kernel on labels, $K_t^\dagger$. The kernel can be constructed in various forms. Without loss of generality, we assume that there are c classes or unique labels, and $\forall i,\ t_i \in \{1, \ldots, c\}$. We choose a kernel of the form $K_t = H(BB^\top)H$, where H is the centering matrix and B is defined as:

$$B = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_c], \tag{19}$$

where $\boldsymbol{b}_q(j)$ is one if $t_j = q$, and zero otherwise. For example if there are five data points such that the first three data points belong to class 1 and the forth and the fifth data points are from class 2, K_t can be formed as follows:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}. \tag{20}$$

Computing $(K_t)^\dagger$ is very easy, because H is *idempotent* and $H^\dagger = H$, we have $(K_t)^\dagger = H(BB^\top)^\dagger H$. It is easy to show that $BB^\top$ has c eigenvalues of $n_1, \ldots, n_c$ associated with $\boldsymbol{b}_1/\sqrt{n_1}, \ldots, \boldsymbol{b}_c/\sqrt{n_c}$ eigenvectors, respectively, where n_q denotes the number of samples in class q. Therefore, we have:

$$(BB^\top)^\dagger = \sum_{q=1}^{c} \frac{\boldsymbol{b}_q \boldsymbol{b}_q^\top}{n_q^2}. \tag{21}$$

Consequently, $(K_t)^\dagger$ is given by:

$$(K_t)^\dagger = H\left(\sum_{q=1}^{c} \frac{\boldsymbol{b}_q \boldsymbol{b}_q^\top}{n_q^2}\right) H. \tag{22}$$

5.2. Classification

In order to test the performance of GLLE compared to LLE and α-SLLE, we ran a set of classification experiments on several UCI datasets. For each dataset, the data points are first normalized to have zero mean and unit standard deviation in each dimension. Since each algorithm relies on a number of input parameters (number of nearest neighbors k, α in α-SLLE, and γ in GLLE), 5-fold cross-validation is performed on the data to determine the optimal parameter settings.

We used the algorithms on the normalized data to produce two-dimensional embeddings of each dataset. Finally, a linear SVM classifier was trained on a training set, and a test dataset was used to evaluate the classification performance. Note that LLE and GLLE do not provide any direct way to handle out-of-sample (test) examples. A common approach to resolve this problem is to learn a non-parametric model between the low and high-dimensional spaces. In this approach, a high-dimensional test data point $\boldsymbol{x}$ is mapped to the low-dimensional space in three steps: (i) the k nearest neighbors of $\boldsymbol{x}$ among the training inputs (in the original space) are identified; (ii) the linear weights that best reconstruct $\boldsymbol{x}$ from its neighbors, subject to a sum-to-one constraint, are computed; (iii) the low-dimensional representation of $\boldsymbol{x}$ is computed as the weighted sum (with weights computed in the previous step) of the embedded points corresponding to those k neighbors of $\boldsymbol{x}$ in the original space. In all of the examples in this paper, the test set embedding is conducted using this non-parametric model.

Table 1
Classification results of GLLE, α-SLLE, and LLE on a number of datasets; algorithm accuracy is given as a percentile. "num." is the number of points in the dataset, and "dim." is the dimensionality. A SVM classifier with a linear kernel is used for classification. The optimal parameter(s) of different algorithms are shown, as determined by 5-fold cross-validation.

Dataset	num.	dim.	Classes	GLLE	k	γ	α-SLLE	k	α	LLE	k
Protein	116	20	6	64.5	5	0.75	56.2	5	0.1	47.6	90
Housing	506	13	2	93.1	5	0.0	93.1	5	0.0	93.1	5
Wine	178	13	3	99.1	30	0.1	96.6	100	0.01	96.6	100
Balance	625	5	3	94.4	15	0.05	91.7	30	1.0	69.2	100
Ion	351	34	2	92.8	50	0.25	92.2	50	0.25	79.1	5
Soybean	47	35	4	100.0	30	0.05	98.3	35	0.01	98.3	35

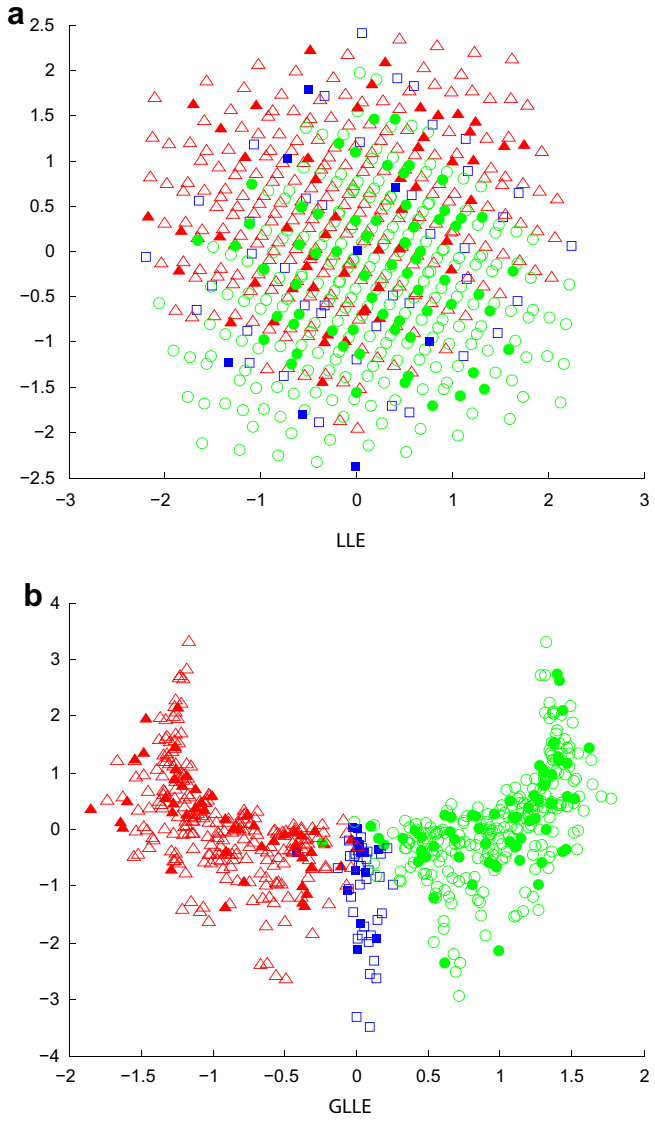

Fig. 1. Comparison of LLE ($k = 100$) with 0.05-GLLE ($k = 15$) for the UCI Balance dataset. Data points of different classes are shown with different symbols. Training data points are hollow and test data points are filled.

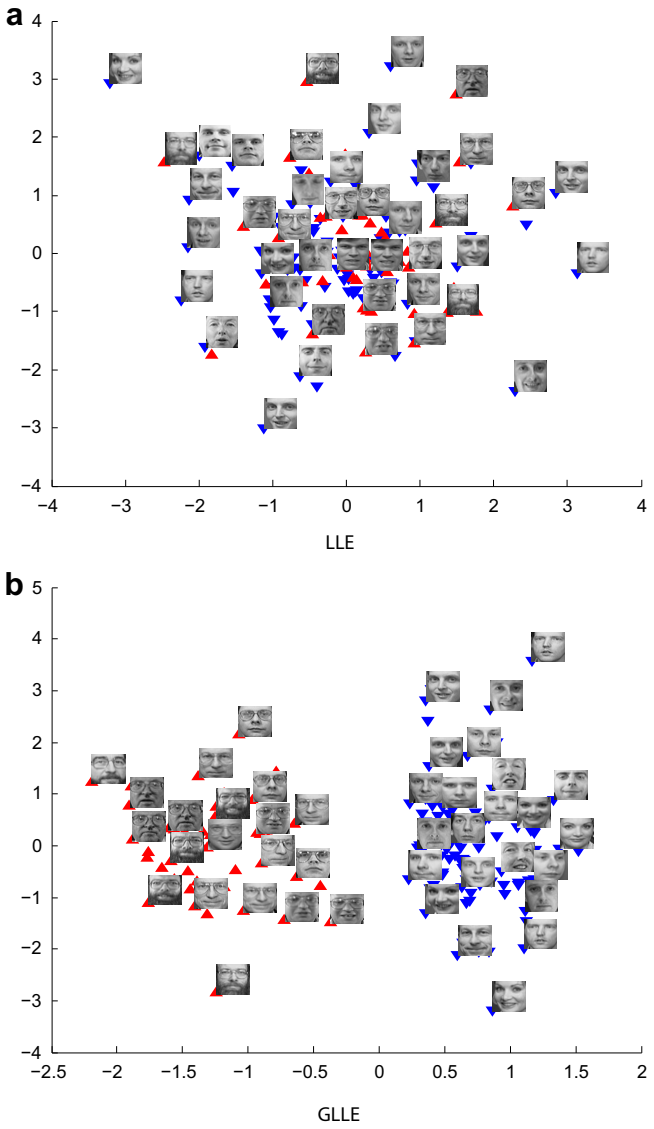

Fig. 2. Comparison of visualizations acquired by LLE and 0.5-GLLE ($k = 50$). There are two groups: persons with and without glasses.

We chose the linear SVM classifier to highlight the linear separability of the different groups in the low-dimensional embeddings. The accuracy results are listed in Table 1.

It is interesting that by using only two dimensions, the performance figures are appealing, except for the Protein dataset which has six classes and 20 dimensions. Furthermore, the optimal number of nearest neighbors k is usually smaller for GLLE, which results in faster computation times. An interesting dataset is Balance, for which applying the SVM on the original data results in 20% better accuracy compared to LLE. As can be seen in Fig. 1(a), LLE's embedded points of different classes severely overlap, while for GLLE the samples are clearly separated (see Fig. 1(b)).

5.3. Visualization

In this section, we compare the embeddings that are generated by LLE and GLLE for data visualization. GLLE uses target variables

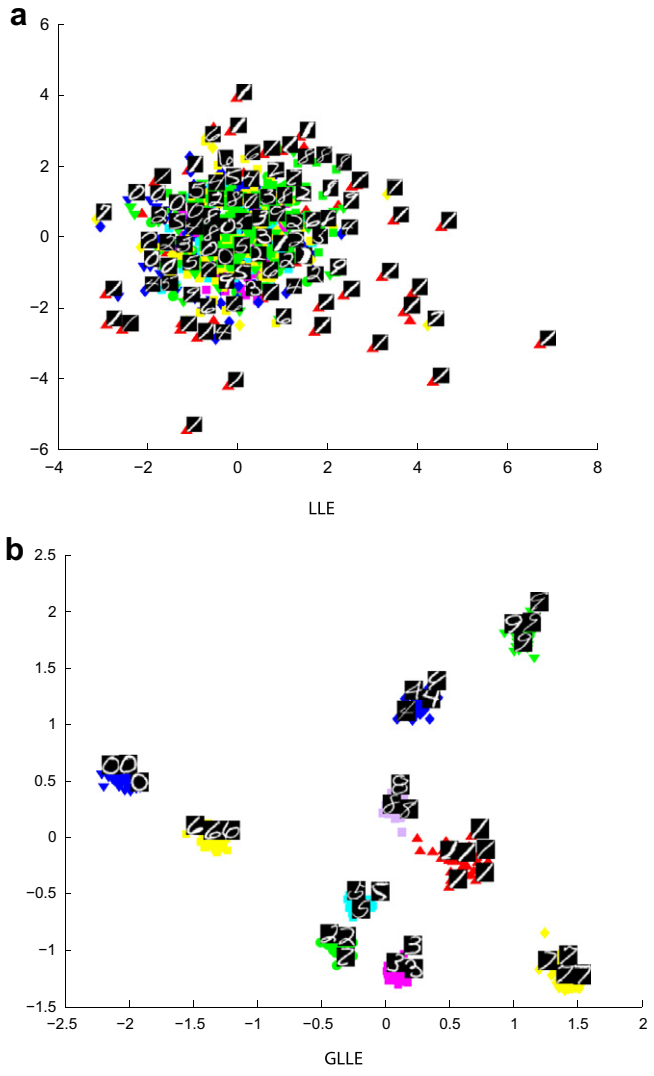

Fig. 3. Comparison of visualizations of the embeddings acquired by LLE and 0.75-GLLE ($k = 50$). The figures are generated by randomly sampling 1000 points from the USPS handwritten digits dataset, and using each algorithm to compute the low-dimensional embedding.

to guide the embeddings about some characteristic of interest in each dataset. The embeddings can then be examined to see whether they capture that characteristic. To show that the target variable truly guides the embedding, three data sets are examined.

Fig. 1 demonstrates results on the UCI Balance data set, which consists of three classes of categorical data in five dimensions. The two-dimensional embedding computed by LLE scatters data from each class across the entire manifold. Alternatively, GLLE yields an embedding where two of the classes are mostly polarized, with one class in-between. This visualization qualitatively demonstrates the effectiveness of GLLE and the impact of the target variable information.

The next two experiments are conducted on very high-dimensional image datasets. One of these experiments is performed on 200 examples of 4096-dimensional Olivetti Faces and the other one on 1000 examples of 256-dimensional USPS handwritten digits.

The results of the Olivetti Faces, shown in Fig. 2, demonstrate the benefit of GLLE on a high-dimensional image dataset. There are 200 images of faces and one distinction is identified by labels: faces with glasses vs. faces without glasses.

In this case, again, a two-dimensional embedding is shown. The two classes are marked with different markers. Additionally, a subset of the images are displayed on the plot (including all images renders the plot unreadable).

Note that, in general, GLLE manages to separate the data based on the target property, whereas the original LLE is typically chaotic.

Finally, Fig. 3 compares LLE and GLLE on 1000 randomly selected samples from the USPS handwritten digits dataset. All ten groups of the digits are visualized, with GLLE separating the classes into disjoint groups. Furthermore, classes that are visually similar located closer together in the two-dimensional embedding.

6. Conclusion

We have proposed a novel, supervised extension to Locally Linear Embedding (LLE). The proposed method, Guided LLE (GLLE), is inspired by the Hilbert–Schmidt Independence Criterion and can make use of many types of side-information represented in the form of target variables. The algorithm has some interesting advantages vs. other supervised extensions of LLE, and we have demonstrated its effectiveness in classification and data visualization tasks.

References

Alipanahi, B., Biggs, M., Ghodsi, A., 2008. Distance metric learning versus fisher discriminant analysis. In: Proc. 23rd AAAI Conf. on Artificial Intelligence, pp. 598–603.

Basu, S., Bilenko, M., Mooney, R.J., 2004. A probabilistic framework for semi-supervised clustering. KDD, 59–68.

Bengio, Y., Paiement, J.-F., 2003. Learning eigenfunctions of similarity: Linking spectral clustering and Kernel PCA. Tech. Rep. 1232, Universite de Montreal (P.V.).

Bilenko, M., Basu, S., Mooney, R.J., 2004. Integrating constraints and metric learning in semi-supervised clustering. In: ICML '04: Proc. 21st Internat. Conf. on Machine Learning. ACM International Conference Proceeding Series. ACM, p. 11+.

Chang, H., Yeung, D.-Y., 2004. Locally linear metric adaptation for semi-supervised clustering. In: ICML '04: Proc. 21st Internat. Conf. on Machine Learning. ACM International Conference Proceeding Series. ACM, pp. 153–160.

Chang, H., Yeung, D.-Y., 2006. Locally linear metric adaptation with application to semi-supervised clustering and image retrieval. Pattern Recognition 39 (7), 1253–1264.

Cook, R.D., Weisberg, S., 1991. Discussion of Li (1991). J. Amer. Statist. Assoc. 86, 328–332.

Cook, R.D., Yin, X., 2001. Dimension reduction and visualization in discriminant analysis (with discussion). Aust. N.Z. J. Statist. 43, 147–199.

de Ridder, D., Kouropteva, O., Okun, O., Pietikäinen, M., Duin, R., 2003. Supervised locally linear embedding. In: Artificial Neural Networks and Neural Information Processing ICANN/ICONIP 2003, p. 175.

Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Ann. Eugen. 7, 179–188.

Fukumizu, K., Bach, F.R., Jordan, M.I., 2004. Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. J. Machine Learn. Res. 5, 73–99.

Globerson, A., Roweis, S.T., 2006. Metric learning by collapsing classes. Adv. Neural Inform. Process. Systems 18, 451–458.

Gretton, A., Bousquet, O., Smola, A., Schölkopf, B., 2005. Measuring statistical dependence with Hilbert–Schmidt norms. In: Algorithmic Learning Theory, 16th Internat. Conf., ALT 2005, Singapore, October 2005, Proc.. Lecture Notes in Artificial Intelligence, vol. 3734. Springer, pp. 63–77.

Gretton, A., Fukumizu, K., Teo, C.H., Song, L., Schölkopf, B., Smola, A., 2008. A kernel statistical test of independence. In: Platt, J., Koller, D., Singer, Y., Roweis, S. (Eds.), Advances in Neural Information Processing Systems, vol. 20. MIT Press, Cambridge, MA, pp. 585–592.

Ham, J., Lee, D., Mika, S., Schölkopf, B., 2004. A kernel view of the dimensionality reduction of manifolds. In: Proc. 21st Internat. Conf. on Machine Learning, p. 47+.

Hristache, M., Juditsky, A., Polzehl, J., Spokoiny, V., 2001. Structure adaptive approach for dimension reduction. Ann. Statist. 29, 1537–1566.

Kouropteva, O., Okun, O., Pietikäinen, M., 2003. Supervised locally linear embedding algorithm for pattern recognition. In: IbPRIA, pp. 386–394.

Li, K., 1991. Sliced inverse regression for dimension reduction (with discussion). J. Amer. Statist. Assoc. 86, 316–342.

Li, K., 1992. On principal hessian directions for data visualization and dimension reduction: Another application of stein's lemma. J. Amer. Statist. Assoc. 87, 1025–1039.

Li, B., Zheng, C., Huang, D., 2008. Locally linear discriminant embedding: An efficient method for face recognition. Pattern Recognition 41 (12), 3813–3821.
Roweis, S., Saul, L., 2000. Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326.
Samarov, A.M., 1993. Exploring regression structure using nonparametric functional estimation. J. Amer. Statist. Assoc. 88, 836–847.
Saul, L., Roweis, S., 2003. Think globally, fit locally: Unsupervised learning of nonlinear manifolds. J. Machine Learn. Res. 4, 119–155.
Schölkopf, B., Smola, A., 2002. Learning with Kernels. MIT Press, Cambridge, MA.
Torkkola, K., 2003. Feature extraction by non-parametric mutual information maximization. J. Machine Learn. Res. 3, 1415–1438.
Weinberger, K.Q., Blitzer, J., Saul, L.K., 2006. Distance metric learning for large margin nearest neighbor classification. Adv. Neural Inform. Process. Systems 18, 1473–1480.
Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S., 2002. Distance metric learning with application to clustering with side-information. Adv. Neural Inform. Process. Systems 15, 505–512.
Yeung, D.-Y., Chang, H., 2006. Extending the relevant component analysis algorithm for metric learning using both positive and negative equivalence constraints. Pattern Recognition 39 (5), 1007–1010.
Zhang, S., 2009. Enhanced supervised locally linear embedding. Pattern Recognition Lett. 30 (13), 1208–1218.
Zhang, Z., Zhao, L., 2007. Probability-based locally linear embedding for classification. In: FSKD '07: Proc. Fourth Internat. Conf. on Fuzzy Systems and Knowledge Discovery. IEEE Computer Society, pp. 243–247.
Zhao, Q., Zhang, D., Lu, H., 2005. Supervised LLE in ICA space for facial expression recognition. In: ICNN&B '05: Internat. Conf. on Neural Networks and Brain, vol. 3, pp. 1970–1975.