Learning Locality-Preserving Discriminative Features

Chang Wang and Sridhar Mahadevan

Computer Science Department
University of Massachusetts Amherst
Amherst, Massachusetts 01003
{chwang, mahadeva}@cs.umass.edu

Abstract. This paper describes a novel framework for learning discriminative features, where both labeled and unlabeled data are used to map the data instances to a lower dimensional space, preserving both class separability and data manifold topology. In contrast to linear discriminant analysis (LDA) and its variants (like semi-supervised discriminant analysis), which can only return $c-1$ features for a problem with c classes, the proposed approach can generate d features, where d is bounded only by the dimensionality of the original problem. The proposed framework can be used with both two class and multiple class problems. It can also be adapted to problems where class labels are continuous. We describe and evaluate the new approach both theoretically and experimentally, and compare its performance with other state of the art methods.

Key words: Feature Selection, Extraction, and Construction, Manifold Learning, Semi-supervised Learning

1 Introduction

In many areas of data mining and information retrieval, it is highly desirable to map high dimensional data instances to a lower dimensional space, preserving topology of the given data manifold. In this paper, we consider a more general problem: learning lower dimensional embedding of data instances preserving both manifold topology and discriminative information to separate instances from different classes. Our proposed approach has its goal to eliminate useless features and improve the speed and performance of classification, clustering, ranking, and multi-task learning algorithms. Our work is related to previous work on regression models, manifold regularization [1], linear discriminant analysis (LDA) [2], and dimensionality reduction methods such as locality-preserving projections (LPP) [3].

Linear regression involves estimating a coefficient vector of dimensionality equal to the number of input features using which data instances are mapped to real-valued outputs (or continuous class labels). For example, given a set of instances $\{x_i\}$ defined in a p dimensional space, a linear regression model computes $\beta_0, \cdots, \beta_p$ such that label y_i can be approximated by $\hat{y}_i = \beta_0 + \beta_1 x_i(1) +$

$\cdots + \beta_p x_i(p)$ for $i = 1, \ldots, n$. The framework of manifold regularization [1] combines the standard loss functions associated with regression or classification with an additional term that preserves the local geometry of the given data manifold (the framework has another term corresponding to an ambient regularizer). One problem solved under this framework can be characterized as follows: given an input data set $X = (x_1, \cdots, x_m)$ and label information $V = (v_1, \cdots, v_l)$ $(l \leq m)$, we want to compute a function f that maps x_i to a new space, where $f^T x_i$ matches x_i's label y_i. In addition, we also want f to preserve the neighborhood relationship within data set X (making use of both labeled and unlabeled data). This problem can be viewed as finding an f that minimizes the cost function: $C(f) = \sum_{i \leq l}(f^T x_i - y_i)^2 + \mu \sum_{i,j}(f^T x_i - f^T x_j)^2 W_X(i,j)$. We can interpret the first mean-squared error term of $C(f)$ as penalizing the difference between a one-dimensional projection of the instance x_i and the label y_i. The second term enforces the preservation of the neighborhood relationship within X (where W_X is a similarity measure). Under this interpretation, manifold regularization constructs embeddings preserving both the topology of the manifold and a 1-dimensional real-valued output structure. The proposed approach generalizes this idea to compute higher order locality-preserving discriminative projections, for both discrete as well as continuous-valued labels.

Linear Discriminant Analysis (LDA) and some of its extensions like semi-supervised discriminant analysis [4, 5] find a dimensionality-reducing projection that best separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or for dimensionality reduction before later classification. However, for a data set with c class labels, LDA type approaches can only achieve a $c-1$ dimensional embedding (since the matrix to model the between-class difference only has $c-1$ nontrivial eigenvectors). In many applications, $c-1$ is far from sufficient. For example, given a data set with two class labels (positive/negative), LDA type approaches only yield a 1D embedding for each instance, even when the data is defined by several hundreds of features in the original space.

Many linear (e.g. PCA) and nonlinear (e.g. Laplacian eigenmaps [6]) dimensionality reduction methods convert dimensionality reduction problems to an eigenvalue decomposition. One key limitation of these approaches is that when they learn lower dimensional embeddings, they do not take label information into account. So only the information that is useful to preserve the topology of the whole manifold is guaranteed to be kept, and the discriminative information separating instances from different classes may be lost. For example, when we are required to describe a human being with a couple of words, we may use such characteristics as two eyes, two hands, two legs and so on. However, none of these features is useful to separate men from women. Similar to our approach, the well-known Canonical Correlation Analysis (CCA) also simultaneously computes two mapping functions. CCA finds linear functions that map instances from two different sets to one space, where the correlation between the corresponding points is maximized. There are two fundamental differences between our approach and CCA: 1. The number of non-zero solutions to CCA is limited to the smallest dimensionality of the input data. For our case, CCA can only get a $c-1$ dimensional embedding since the label is in a c dimensional space. 2. Our approach can make use of unlabeled data, while CCA cannot.

The proposed approach can be distinguished from some recent work. LDPP [7] learns the dimensionality reduction and nearest neighbor classifier parameters jointly. LDPP does not preserve the topology of the given data set. The algorithm in [8] provides a framework to learn a (local optimal) linear mapping function to map the given data to a new space to enhance a given classifier. Their mapping function is designed for classification only and does not preserve the topology of the data set. Transductive component analysis [9] can return a lower dimensional embedding of an arbitrary dimensionality, preserving manifold topology. It differs from our approach in that our approach can be adapted to handle continuous labels. Colored maximum variance unfolding is not directly related to our work either. It is designed to preserve local distance structure [10].

In this paper, we develop a framework for learning optimal discriminative projections to map high-dimensional data instances to a new lower dimensional space, leveraging the given class label information such that instances from different classes will be mapped to different locations. Similar to the goal of manifold-preserving dimensionality reduction approaches, we also want the topology of the given data manifold to be respected. Our new approach combines the ideas of manifold regularization, LDA and regular dimensionality reduction. Both LDA and our approach provide discriminative projections to separate instances from different classes, but LDA can only return $c-1$ dimensional projections for a problem with c classes. Compared to dimensionality reduction methods like PCA, our approach preserves both manifold topology and class separability. In addition to the benefits discussed above, our approach has an added benefit that it can be extended to handle continuous class labels. In many applications, we might not have discrete class labels. Instead, each instance is assigned with a value and two instances are supposed to be similar if their associated values are similar. Such values can be treated as continuous (real-valued) labels. The ability to handle continuous labels is also useful in the other scenarios. For example, we might use 4 values to label the instances in a ranking algorithm: 1-“excellent”, 2-“good”, 3-“fair”, 4-“bad”. The instances labeled with “excellent” should be more similar to the instances labeled with “good”, compared to the instances labeled with “bad”.

The rest of this paper is organized as follows. In Section 2 we describe our algorithm to address regular two class/multiple class problems. In Section 3, we show how our algorithm can be extended to solve problems with continuous labels. Section 4 summarizes our experimental results. Section 5 provides some concluding remarks.

2 Overall Framework

We introduce the overall framework in this section. It is helpful to review the notation described below. In particular, we assume that class labels can be viewed as c-dimensional real-valued vectors if there are c possible labels.

2.1 The Problem

Assume the given data set $X = (x_1, \cdots, x_m)$ is a $p \times m$ matrix, where instance x_i is defined by p features. $c =$ number of classes in X. Label y_i is a $c \times 1$ vector representing x_i's class label. If x_i is from the j^{th} class, then $y_i(j) = 1$; $y_i(k) = 0$ for any $k \neq j$. We also assume x_i's label is given as y_i for $1 \leq i \leq l$; x_i's label is not available for $l+1 \leq i \leq m$. $Y = (y_1, \cdots, y_l)$ is a $c \times l$ matrix.

The problem is to compute mapping functions f (for data instances) and g (for labels) to map data instance $x_i \in R^p$ and label $y_i \in R^c$ to the same d-dimensional space, where the topology of the data manifold is preserved, the instances from different classes are separated and $d \ll p$. Here, f is a $p \times d$ matrix and g is a $c \times d$ matrix.

2.2 The Cost Function

The solution to the overall problem of learning locality preserving discriminative projections can be formulated as constructing mapping functions f and g that minimize the cost function

$$C(f,g) = \frac{\sum_{i \leq l} \|f^T x_i - g^T y_i\|_2^2 + \mu \sum_{i,j} \|f^T x_i - f^T x_j\|_2^2 W_X(i,j)}{\sum_{i \leq l} \sum_{k=1, s_k \neq y_i}^{c} \|f^T x_i - g^T s_k\|_2^2},$$

where s_k and W_X are defined as follows: s_k is a $c \times 1$ matrix. $s_k(k) = 1$, and $s_k(j) = 0$ for any $j \neq k$. S_k is a $c \times l$ matrix= $(s_k, \cdots, s_k)$. W_X is a matrix, where $W_X(i,j)$ is the similarity (could be defined by heat kernel) between x_i and x_j.

Here, $f^T x_i$ is the mapping result of x_i. $g^T y_i$ (or $g^T s_k$) is the mapping result of label y_i (or s_k). The first term in the numerator represents the difference between the projection result of any instance x_i and its corresponding label y_i. We want this value to be small, since this makes x_i be close to its true label. The second term in the numerator models the topology of data set X using both labeled and unlabeled data. When it is small, it encourages the neighborhood relationship within X to be preserved. μ is a weight to balance the first and second terms. It is obvious that we want the numerator of $C(f,g)$ to be as small as possible. The denominator models the distance between the projection result of each instance x_i and all its wrong labels. We want this value to be as large as possible, since this makes x_i be far away from its wrong labels.

Thus, minimizing $C(f,g)$ will preserve the topology of data set X, and project instances to a new lower dimensional space, where the instances from different classes are well separated from each other.

2.3 High Level Explanation

Manifold regularization addresses the problem of learning projections to map the data instances (with known labels) to their class labels, preserving the manifold topology. The loss function used in one algorithm under the manifold regularization framework is as follows:

$$C(f) = \sum_{i \leq l} (f^T x_i - y_i)^2 + \mu \sum_{i,j} (f^T x_i - f^T x_j)^2 W_X(i,j),$$

where y_i is the real-valued label of x_i. This loss function can be relaxed for our problem, since our goal is to separate instances from different classes. It is less important whether the embedding of each instance is close to its given class label or not. In our algorithm, we have a mapping function f for data instances, and g for labels such that f and g can work together to map the data instances and labels to the same space, where the mapping results of instances and their labels are close to each other. The mapping g allows us to scale the entries of the label vector by different amounts, which then allows better projections of points. An illustration of this idea is given by Figure 1.

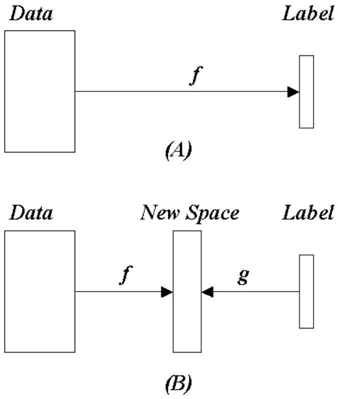

Fig. 1. Illustration of regular regression approaches (A), and our approach (B).

In summary, the numerator of our loss function encourages the instances with the same label to stay together, preserving the data manifold topology. The denominator of the loss function encourages the instances with different labels to be away from each other.

2.4 Discriminative Projections: The Main Algorithm

Some notation used in the algorithm is as follows:
$\gamma = (f^T, g^T)^T$ is a $(p + c) \times d$ matrix. $Tr()$ means trace. I is an $l \times l$ identity

matrix. $U_1 = \begin{pmatrix} I\, 0 \\ 0\, 0 \end{pmatrix}_{m\times m}$, $U_2 = U_3^T = \begin{pmatrix} I \\ 0 \end{pmatrix}_{m\times l}$, $U_4 = I$.

The algorithmic procedure is as follows:

1. **Construct matrices A, B and C:**

$$A = \begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix} \begin{pmatrix} U_1 & -U_2 \\ -U_3 & U_4 \end{pmatrix} \begin{pmatrix} X^T & 0 \\ 0 & Y^T \end{pmatrix}$$

$$B = \sum_{k=1}^{c} \begin{pmatrix} X & 0 \\ 0 & S_k \end{pmatrix} \begin{pmatrix} U_1 & -U_2 \\ -U_3 & U_4 \end{pmatrix} \begin{pmatrix} X^T & 0 \\ 0 & S_k^T \end{pmatrix}$$

$$C = \begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix} \begin{pmatrix} \mu L_x & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} X^T & 0 \\ 0 & Y^T \end{pmatrix}$$

2. **Compute $\gamma = (\gamma_1, \cdots, \gamma_d)$: the d minimum eigenvector solutions to the generalized eigenvalue decomposition equation:**

$$(A + C)x = \lambda(B + C)x.$$

3. **Compute discriminative projection functions f and g:**
$\gamma = (\gamma_1, \cdots, \gamma_d)$ is a $(p+c) \times d$ matrix, whose top p rows= mapping function f, the next c rows= mapping function g. i.e.

$$\begin{pmatrix} f \\ g \end{pmatrix} = (\gamma_1, \cdots, \gamma_d).$$

4. **Compute the d-dimensional embedding of data set X:**
The d-dimensional embedding of X is $f^T X$, whose i^{th} column represents the embedding of x_i.

2.5 Justification

Theorem 1: d minimum eigenvector solutions to $(A + C)x = \lambda(B + C)x$ provide the optimal d-dimensional discriminative projections to minimize the cost function $C(f, g)$.

Proof: Given the input and the cost function, the problem is formalized as:

$$\{f, g\} = \arg_{f,g} \min(C(f, g)).$$

When $d = 1$, we define M, N and L as follows:

$$M = \sum_{i\leq l} (f^T x_i - g^T y_i)^2, N = \sum_{i\leq l} \sum_{k=1}^{c} (f^T x_i - g^T s_k)^2,$$

$$L = \mu \sum_{i,j} (f^T x_i - f^T x_j)^2 W_X(i, j).$$

It is easy to verify that:

$$\begin{aligned}\arg_{f,g}\min(C(f,g)) &= \arg_{f,g}\min\frac{M+L}{N-M} = \arg_{f,g}\max\frac{N-M}{M+L}\\ &= \arg_{f,g}\max\frac{N-M+M+L}{M+L} = \arg_{f,g}\max\frac{N+L}{M+L}\\ &= \arg_{f,g}\min\frac{M+L}{N+L}.\end{aligned}$$

$$M = \sum_{i\le l}(f^Tx_i - g^Ty_i)^2 = (f^TX, g^TY)\begin{pmatrix} U_1 & -U_2\\ -U_3 & U_4\end{pmatrix}\begin{pmatrix} X^Tf\\ Y^Tg\end{pmatrix} = \gamma^TA\gamma.$$

$$N = \sum_{i\le l}\sum_{k=1}^{c}(f^Tx_i - g^Ts_k)^2 = (f^T, g^T)B\begin{pmatrix} f\\ g\end{pmatrix} = \gamma^TB\gamma.$$

$$L = \mu\sum_{i,j}(f^Tx_i - f^Tx_j)^2W_X(i,j) = \mu f^TXL_XX^Tf = \gamma^TC\gamma.$$

So

$$\arg_{f,g}\min C(f,g) = \arg_{f,g}\min\frac{M+L}{N+L} = \arg_{f,g}\min\frac{\gamma^T(A+C)\gamma}{\gamma^T(B+C)\gamma}.$$

It follows directly from the Lagrange multiplier method that the optimal solution that minimizes the loss function $C(f,g)$ is given by the minimum eigenvector solution to the generalized eigenvalue problem:

$$(A+C)x = \lambda(B+C)x.$$

When $d > 1$,

$$M = \sum_{i\le l}\|f^Tx_i - g^Ty_i\|_2^2 = Tr((\gamma_1\cdots\gamma_d)^TA(\gamma_1\cdots\gamma_d)).$$

$$N = \sum_{i\le l}\sum_{k=1}^{c}\|f^Tx_i - g^Ts_k\|_2^2 = Tr((\gamma_1\cdots\gamma_d)^TB(\gamma_1\cdots\gamma_d)).$$

$$L = \mu\sum_{i,j}\|f^Tx_i - f^Tx_j\|_2^2W_X(i,j) = Tr((\gamma_1\cdots\gamma_d)^TC(\gamma_1\cdots\gamma_d)).$$

$$\arg_{f,g}\min C(f,g) = \arg_{f,g}\min\frac{Tr((\gamma_1\cdots\gamma_d)^T(A+C)(\gamma_1\cdots\gamma_d))}{Tr((\gamma_1\cdots\gamma_d)^T(B+C)(\gamma_1\cdots\gamma_d))}.$$

Standard approaches [11] show that the solution to $\gamma_1\cdots\gamma_d$ that minimizes $C(f,g)$ is provided by the eigenvectors corresponding to the d lowest eigenvalues of the generalized eigenvalue decomposition equation:

$$(A+C)x = \lambda(B+C)x.$$

□

3 Extension to Continuous Labels

3.1 The Problem

The algorithm discussed in the previous section can handle both two class and multiple class problems. However, in many applications, we might not have class label information. Instead, each instance x_i is assigned with a value and x_i and x_j are supposed to be similar if their associated values are similar. This value can be treated as a continuous (real-valued) label associated with each instance. We call this problem a problem with continuous labels. One example of this is from reinforcement learning, where each (high-dimensional) state in the state space of a Markov decision process is assigned with a value to represent its long-term reward achieved starting from that state.

The ability to handle continuous labels is useful in many applications like ranking, where the instances with similar labels are supposed to be similar. For example, a search engine needs to rank many documents for each query. The documents that are ranked at the top of the result list will be returned to the users as the retrieval results. In this scenario, the documents labeled with "perfect match" should be more similar to the documents labeled with "good match", compared to the documents labeled with "bad match". Such a problem cannot be solved by approaches like semi-supervised discriminant analysis [4] and transductive component analysis [9], which can not handle continuous labels.

The problem to solve in this section is described as follows: given a data set $X = (x_1, \cdots, x_m)$ and real-valued label information $Y = (y_1, \cdots, y_l)$ $(l \leq m)$, we compute mapping functions f (for data instances) and g (for labels) to map the data instances $x_i \in R^p$ and the real-valued labels y_i to the same d-dimensional space, where the topology of the data manifold is preserved, and the instances with similar labels are mapped to the similar locations.

3.2 The Cost Function

The new cost function

$$C(f, g) = \sum_{i \leq l} \|f^T x_i - g^T y_i\|_2^2 + \mu \sum_{i,j} \|f^T x_i - f^T x_j\|_2^2 W_X(i, j),$$

where g re-scales label y_i. To remove an arbitrary scaling factor in the embedding, we impose an extra constraint: $f^T X D_X X^T f + g^T Y Y^T g = I$. It is easy to verify that $\|y_i - y_j\|_2 \leq \|y_i - y_k\|_2 \to \|g^T y_i - g^T y_j\|_2 \leq \|g^T y_i - g^T y_k\|_2$. This property guarantees that similar labels will be mapped to similar locations in the new space.

In $C(f, g)$, the first term penalizes the difference between the projection result of x_i and the corresponding real-valued label y_i. The second term encourages the neighborhood relationship within X to be preserved. Compared to the cost function discussed in the previous section, the new cost function does not have the denominator part. The reason for this is when labels are continuous, we have an unlimited number of possible labels. So we cannot construct the denominator part as before, and instead instances with different labels are projected away from each other by mapping them to their labels.

3.3 Discriminative Projections over Continuous Labels

In this section, we discuss how to adapt the algorithm in the previous section to the problems with continuous labels. In the new settings, $Y = (y_1, \cdots, y_l)$ becomes a $1 \times l$ matrix, where y_i represents the real-valued label assigned to x_i. We define five new matrices as follows:
D_X is a diagonal matrix: $D_X(i,i) = \sum_j W_X(i,j)$.
$L_X = D_X - W_X$ is the graph Laplacian matrix corresponding to W_X.
$Z = \begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix}$ is a $(p+1) \times (m+l)$ matrix.
$\hat{L} = \begin{pmatrix} U_1 + \mu L_X & -U_2 \\ -U_3 & U_4 \end{pmatrix}$ is an $(m+l) \times (m+l)$ matrix.
$\hat{D} = \begin{pmatrix} D_X & 0 \\ 0 & I \end{pmatrix}$ is also an $(m+l) \times (m+l)$ matrix.

The solution to minimize $C(f,g)$ is given by the minimum eigenvector solution to $Z\hat{L}Z^T x = \lambda Z\hat{D}Z^T x$.

3.4 Justification

Theorem 2: d minimum eigenvector solutions to $Z\hat{L}Z^T x = \lambda Z\hat{D}Z^T x$ provide d-dimensional discriminative projections to minimize $C(f,g)$.
Proof:
Given the input, we want to find the optimal mapping functions f and g such that $C(f,g)$ is minimized:

$$\{f, g\} = \arg_{f,g} \min(C(f,g)).$$

When $d = 1$,
The first term of $C(f,g)$ becomes

$$\sum_{i \le l} (f^T x_i - g^T y_i)^2 = (f^T X, g^T Y) \begin{pmatrix} U_1 & -U_2 \\ -U_3 & U_4 \end{pmatrix} \begin{pmatrix} X^T f \\ Y^T g \end{pmatrix}.$$

The second term can be written as:

$$\mu \sum_{i,j} (f^T x_i - f^T x_j)^2 W_X(i,j) = \mu f^T X L_X X^T f.$$

So

$$\begin{aligned} C(f,g) &= \sum_{i \le l} (f^T x_i - g^T y_i)^2 + \mu \sum_{i,j} (f^T x_i - f^T x_j)^2 W_X(i,j) \\ &= (f^T X, g^T Y) \begin{pmatrix} U_1 + \mu L_X & -U_2 + 0 \\ -U_3 + 0 & U_4 + 0 \end{pmatrix} \begin{pmatrix} X^T f \\ Y^T g \end{pmatrix} = \gamma^T Z\hat{L}Z^T \gamma. \end{aligned}$$

To remove an arbitrary scaling factor in the embedding, we impose an extra constraint:

$$f^T X D_X X^T f + g^T Y Y^T g = \gamma^T Z \hat{D} Z^T \gamma = 1.$$

This constraint balances f and g. Without this constraint, all instances and labels could be mapped to the same location in the new space. Here, the matrix D_X provides a natural measure on the vertices (instances) of the graph. If the value of $D_X(i,i)$ is large, it means x_i is important. Then the problem of minimizing $C(f,g)$ can be written as:

$$\arg \min_{f,g:\gamma^T Z \hat{D} Z^T \gamma = 1} \gamma^T Z \hat{L} Z^T \gamma.$$

The Lagrangian multiplier method shows that the solution to this problem is given by the minimum eigenvector solution to the generalized eigenvalue equation:

$$Z \hat{L} Z^T x = \lambda Z \hat{D} Z^T x.$$

When $d > 1$, the problem of minimizing $C(f,g)$ can be written as:

$$\arg \min_{f,g:\gamma^T Z \hat{D} Z^T \gamma = I} Tr(\gamma^T Z \hat{L} Z^T \gamma).$$

Optimization problems of this type can be solved by standard approaches [2]. The d-dimensional projection is provided by the eigenvectors corresponding to the d lowest eigenvalues of the same generalized eigenvalue decomposition equation. □

In the new settings, $\binom{f}{g} = (\gamma_1, \cdots, \gamma_d)$. One issue that we need to address is how the value of g will be set. Theoretically speaking, g can be very close to a zero vector, which fails to distinguish the difference between different labels. However, this is unlikely to happen in real-world applications. If g is close to zero, the constraint becomes $f^T X D_X X^T f = I$. To minimize the cost function $C(f,g)$, $f^T x_i$ also needs to be close to 0 for $i \in [1,l]$. The new constraint $f^T X D_X X^T f = I$ will prevent that from happening, when the labeled data is well sampled from the original data set.

4 Experimental Results

In this section, we test discriminative projections, manifold regularization, LDA, and LPP using four data sets: recognition of handwritten digits using the USPS dataset (a vision data set with multiple classes), TDT2 data (a text data set with multiple classes), classification of mushrooms (a data set with two classes) and OHSUMED data (a text data set with two classes). We use the following simple strategy to decide the value of μ in the loss function $C(f,g)$. Let $s =$ the sum of all entries of W_X and $l =$ the number of training examples with labels, then l/s balances the scales of the first term and second term in the numerator of $C(f,g)$. We let $\mu = l/s$, if we treat accuracy and topology preservation as equally important. We let $\mu > l/s$, when we focus more on topology preservation;

$\mu < l/s$, when accuracy is more important. In this paper, we use $\mu = l/s$ for regular discriminative projections; $\mu = 0.1 \cdot l/s$ when we assume the label is continuous.

4.1 USPS Digit Data (Vision Data)

The USPS digit data set (http://www.gaussianprocess.org/gpml/data/) has 9,298 images and is randomly divided into a training set (4,649 cases) and a test set (4,649 cases). Each image contains a raster scan of the 16 × 16 grey level pixel intensities. The intensities have been scaled to the range [-1, 1].

We first computed lower dimensional embeddings of the data using regular discriminative projections, LDA and Locality Preserving Projections (LPP). This data set has 10 labels, so LDA can only return an embedding of 9 or less dimensions. LPP and discriminative projections can return an embedding of any dimensionality. The 3D and 2D embedding results are shown in Figure 2, from which we can see that regular discriminative projections and LDA can separate the data instances from different classes in the new space, but LPP can not.

To see how the discriminative information is preserved by different approaches, we ran a leave-one-out test. We first computed 9D and 50D embeddings using discriminative projections and LPP. We also computed 9D embedding using LDA. Then we checked for each point x_i whether at least one point from the same class were among its K nearest neighbors in the new space. We tried $K = 1, \cdots, 10$. The results are summarized in Figure 3. From this figure, we can see that discriminative projections (50 dimensional), (9 dimensional) and LDA (9 dimensional) achieve similar performance, and perform much better than LPP. The results also show that the projections that best preserve the data set topology might be quite different from the projections that best preserve the discriminative information. In Figure 4, we compare discriminative projections (assuming the

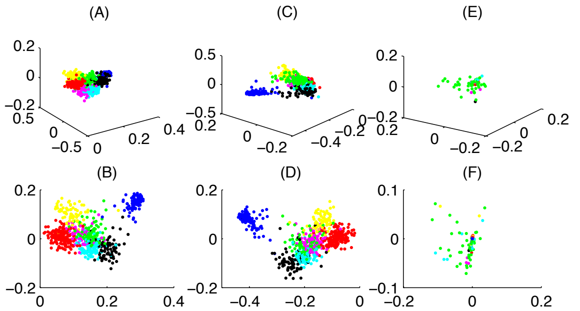

Fig. 2. USPS digit test: (the color represents class label): (A) discriminative projections 3D embedding; (B) discriminative projections 2D embedding; (C) LDA 3D embedding; (D) LDA 2D embedding; (E) LPP 3D embedding; (F) LPP 2D embedding.

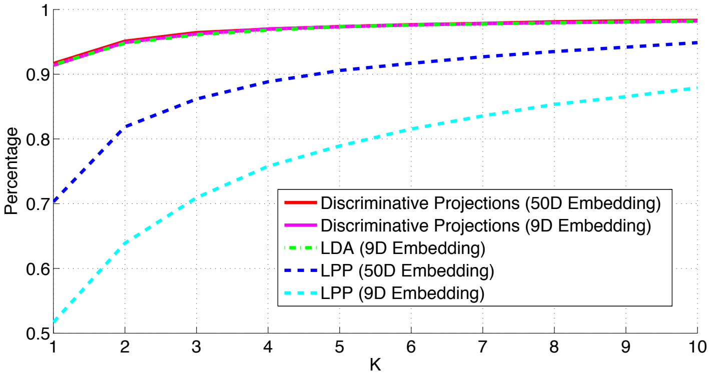

Fig. 3. USPS test: how the discriminative information is preserved.

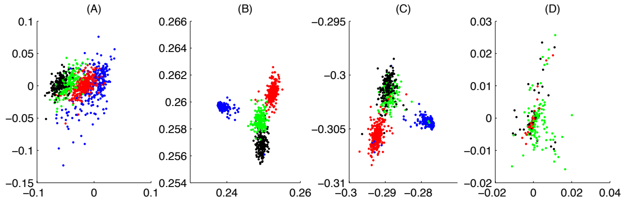

Fig. 4. 2D embedding of USPS digit data (blue: '1', red: '2', green: '3', black: '4'): (A) discriminative projections (assuming the label is continuous); (B) regular discriminative projections; (C) LDA; (D) LPP.

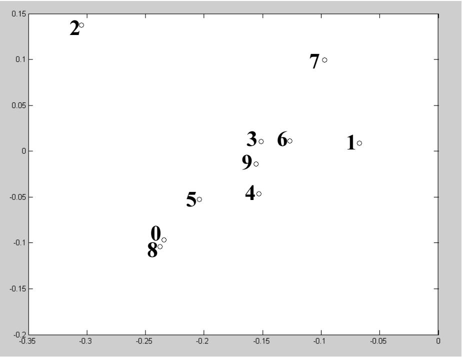

Fig. 5. Projection results of 10 USPS digit labels.

label is continuous) to regular discriminative projections, LDA and LPP using the data for digit '1', '2', '3' and '4'. The reason to select 4 categories was due to the visualization consideration. When we assume the label is continuous, the algorithm treats a label as a value associated with an instance, and the instances that are associated with similar values will be projected to similar locations. Our algorithm is also useful when the class label is discrete (e.g. in this test). In this scenario, instances from different classes are treated differently based on the similarity between classes. From the results, we can see that the new algorithm goes beyond its regular version and LDA in that the instances with similar labels are mapped to similar locations. For example, in the new space, the instances corresponding to digit '1' (label=1) are closer to the instances corresponding to dight '2' (label=2), compared to the instances corresponding to digit '4' (label=4). Regular discriminative projections and LDA treat different classes equally. They make no attempt to map the instances with similar associated values (labels) to the similar locations.

We also used this example to visualize the new "prototype" of each label in a 2D space (Figure 5). The original labels are in a 10D space. The new labels are constructed by projecting the old labels onto the space spanned by the first two columns of mapping function g. From the figure, we can see that new labels of similar digits are close to each other in the new space. For example, '0' and '8' are together; '3', '6' and '9' are also close to each other. This result makes sense, since to preserve local topology of the given data set, similar digits have a large chance of being projected to similar locations. We ran another test with less respect to manifold topology (by setting $\mu = 10^{-10}$). In the new scenario, all 10 new labels were very well separated. This experiment shows that the mapping g allows us to scale the entries of the label vector by different amounts for different applications, which then allows more flexible projections of instances.

4.2 TDT2 Data (Text Data)

The TDT2 corpus consists of data collected during the first half of 1998 and taken from 6 sources, including 2 newswires (APW, NYT), 2 radio programs (VOA, PRI) and 2 television programs (CNN, ABC). It consists of more than 10,000 documents which are classified into 96 semantic categories. In the data set we are using, the documents that appear in more than one category were removed, and only the largest 4 categories were kept, thus leaving us with 5,705 documents in total.

We applied our approach, LDA and LPP to the TDT2 data assuming label information of 1/3 documents from each class was given, i.e. $l = 5,705/3$. We performed a quantitative analysis to see how the topology of the given manifold was preserved. A leave-one-out test was used to compare the lower dimensional embeddings. In this test, we first computed 3D and 100D embeddings using discriminative projections and LPP. We also computed 3D embedding using LDA (recall that LDA can only return embeddings up to 3D for a data set with 4 class labels). Then we checked for each document x_i whether its nearest neighbor in its original space was still among its K neighbors in the new space. We tried

$K = 1, \cdots, 10$. The results are summarized in Figure 6. From this figure, we can see that discriminative projections with 3D embedding, LPP with 3D embedding and LDA are not effective in preserving the manifold topology. It is obvious that 3D embedding is not able to provide sufficient information to model the neighborhood relationship for this test. However, LDA can not go beyond this, since it can only compute embeddings up to 3D for TDT2 data. On the contrary, discriminative projections with 100D embedding and LPP with 100D embedding do a much better job, and the performances of these two approaches are also quite similar.

To see how the discriminative information is preserved by different approaches, we ran a similar leave-one-out test. Again, we first computed 3D and 100D embeddings using both discriminative projections and LPP. We also computed the 3D embedding using LDA. Then we checked for each document x_i whether at least one document from the same class was among its K nearest neighbors in the new space (we use this as correctness). We tried $K = 1, \cdots, 10$. The results are summarized in Figure 7. From this figure, we can see that discriminative projections and LDA perform much better than LPP in all 10 tests. Discriminative projections with 3D embedding and LDA achieve similar results, while discriminative projections with 100D embedding is slightly better.

Generally speaking, LDA does a good job at preserving discriminative information, but it does not preserve the topology of the given manifold and not suitable for many dimensionality reduction applications, which need an embedding defined by more than $c - 1$ features. LPP can preserve the manifold topology, but it totally disregards the label information. Discriminative projections combines both LDA and LPP, such that both manifold topology and the class separability will be preserved. In addition, depending on the applications, users may decide how to choose μ to balance the two goals. If we focus more on the manifold topology, we choose a larger value for μ; otherwise, we choose a smaller value for μ.

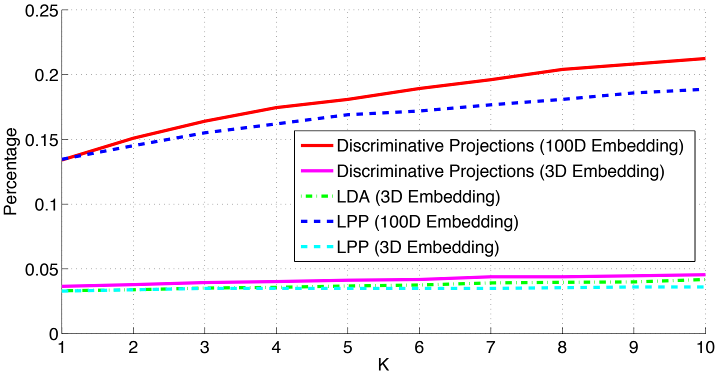

Fig. 6. TDT2 test: how the manifold topology is preserved.

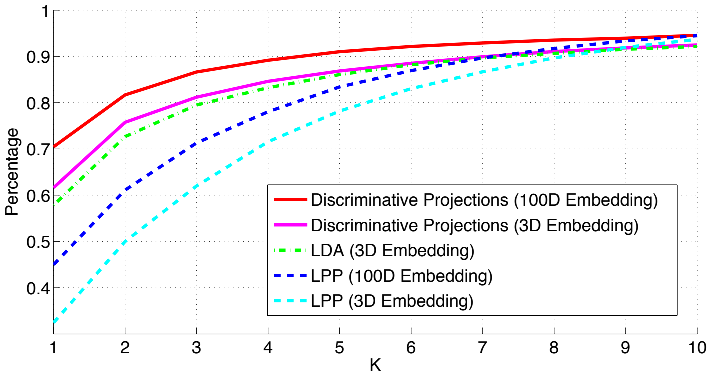

Fig. 7. TDT2 test: how the discriminative information is preserved.

4.3 OHSUMED Data and Mushroom Data

In Section 4.1 and 4.2, we show how discriminative information is preserved. In this section, we directly test how the new approach can improve classification performance. The classifier we are using is a linear regression model. Two datasets are tested in this section: OHSUMED data and Mushroom data.

The OHSUMED collection is a standard dataset provided by the LETOR3.0 collection [12]. The version that we are using contains 63 queries and 10,494 different query-document pairs. There are two labels in the dataset: relevant and non-relevant. All query-document pairs are represented in the same feature space with 45 standard features, including BM25, TF, TF-IDF, etc. In this test, labels of 500 randomly chosen pairs are given and the remaining pairs are held for test. We applied our approach, LDA and LPP to this dataset. LDA can only result in 1D embedding. In discriminative projections and LPP we tested both 1D and 30D embedding. We also ran a test using the original 45 features without doing any feature construction. The Mushroom dataset is a standard dataset from UCI machine learning repository. Mushrooms are described in terms of 112 binary features, and the classification task is to predict whether a mushroom is edible or poisonous. The dataset has 8,124 instances, and 500 instances are used in training. The experiment setting is the same as the OHSUMED test. The results of both tests are summarized in Table 1.

Dataset	Discriminative	Discriminative	LDA	LPP	LPP	Original
	Projections 1D	Projections 30D	1D	1D	30D	
OHSUMED	60.24%	**68.83%**	49.78%	60.73%	64.98%	63.56%
Mushroom data	52.83%	**76.34%**	68.36%	48.95%	52.35%	54.67%

Table 1. Classification Accuracies

The results show that discriminative projections with 30D embeddings improves the classification performances over LDA and LPP in both tests. It is also better than directly using the original features. Discriminative projections fits classification tasks for the following reasons: 1, the mapping g offers us more flexible "labels", which allow better projections of instances; 2, the new discriminative features also take dataset topology into consideration, which lowers the chance of running into overfitting problem; 3, discriminative projections results in more discriminative features than LDA. This is particularly useful to design complicated non-linear classifiers.

5 Conclusions

In this paper, we introduced a novel approach to learn discriminative projections to map high-dimensional data instances to a new lower dimensional space, preserving both manifold topology and class separability. Discriminative projections goes beyond LDA in that it can provide an embedding of an arbitrary dimensionality rather than $c - 1$ for a problem with c class labels. It also differs from regular dimensionality reduction since the discriminative information to separate instances from different classes will be preserved. Our approach is a semi-supervised approach making use of both labeled and unlabeled data. It is general, since it can handle both two class and multiple class problems and can also be adapted to problems with continuous labels. In addition to the theoretical validations, we also presented real-world applications of our approach to information retrieval and computer vision.

References

1. M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 2006.
2. K. Fukunaga. *Introduction to statistical pattern classification.* Academic Press, 1990.
3. X. He and P. Niyogi. Locality preserving projections. In *Proceedings of the Advances in Neural Information Processing Systems*, 2003.
4. D. Cai, X. He, and J. Han. Semi-supervised discriminant analysis. In *International Conference on Computer Vision*, 2007.
5. D. Zhao, Z. Lin, R. Xiao, and X. Tang. Linear laplacian discrimination for feature extraction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
6. M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15, 2003.
7. M. Villegas and R. Paredes. Simultaneous learning of a discriminative projection and prototypes for nearest-neighbor classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
8. D. S. Pham and S. Venkatesh. Robust learning of discriminative projection for multicategory classification on the stiefel manifold. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

9. W. Liu, D. Tao, and J. Liu. Transductive component analysis. In *International Conference on Data Mining*, 2008.
10. L. Song, A. Smola, K. Borgwardt, and A. Gretton. Colored maximum variance unfolding. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.
11. S. S. Wilks. *Mathematical statistics*. Wiley, 1963.
12. Tie-Yan Liu, Tao Qin, Jun Xu, Xiong Wenying, and Hang Li. Letor: Benchmark dataset for research on learning to rank for information retrieval.