Semi-supervised dimensionality reduction for image retrieval

Bin Zhang\textsuperscript{a} Yangqiu Song\textsuperscript{b} Wenjun Yin\textsuperscript{a} Ming Xie\textsuperscript{a} Jin Dong\textsuperscript{a} and Changshui Zhang\textsuperscript{b}

\textsuperscript{a}IBM China Research Lab, Beijing, China; \textsuperscript{b}Tsinghua University, Beijing, China

\textbf{ABSTRACT}

This paper proposes a novel semi-supervised dimensionality reduction learning algorithm for the ranking problem. Generally, we do not make the assumption of existence of classes and do not want to find the classification boundaries. Instead, we only assume that the data point cloud can construct a graph which describes the manifold structure, and there are multiple concepts on different parts of the manifold. By maximizing the distance between different concepts and simultaneously preserving the local structure on the manifold, the learned metric can indeed give good ranking results. Moreover, based on the theoretical analysis of the relationship between graph Laplacian and manifold Laplace-Beltrami operator, we develop an online learning algorithm that can incrementally learn the unlabeled data.

\textbf{Keywords:} Dimensionality Reduction, Ranking, Image Retrieval

\section{1. INTRODUCTION}

Dimensionality reduction techniques are interesting topics in computer vision field, since many of the applications have encountered the high-dimension problem. One of the applications is the appearance-based image recognition problem.\cite{1-4} An image can be represented as a point in very high-dimensional space. By embedding high-dimensional data samples in a low-dimensional space, recognition using the reduced features can be much faster and more robust.\cite{2} In the real world, a great many images can be obtained from the Internet, or from a digital camera for surveillance or web chatting. Labeling all of them is both time consuming and costly. Thus, a semi-supervised technique should be developed. There have been many approaches which can do classification and clustering in a semi-supervised way.\cite{5}

Due to the intrinsic difference between ranking and classification or clustering, we can not make the strong assumptions that the data can be categorized into several classes. In ranking problem, there is no clear classification boundary in the data. Thus, simply labeling may not lead to good ranking results. Therefore, traditional methods, such as Principal Component Analysis (PCA)\cite{6} and Linear Discriminant Analysis (LDA),\cite{7} are not suitable. PCA assumes that data construct one Gaussian cluster and maximizes the data variance, which is equivalent to minimizing the reconstruction error. LDA assumes that each class is a Gaussian and each Gaussian has the same “shape”. It tries to find the transformation that minimizes the within-class scatter and maximizes the between-class scatter simultaneously.

In this paper, we design an algorithm that can deal with ranking tasks. Both the human hints and the geometric information provided by observed data should be used. The essence of the proposed algorithm is a semi-supervised learning method. However, it is different from the start point of traditional semi-supervised classification and clustering.\cite{5} Generally, we do not make the assumption of existence of classes and do not want to find the classification boundaries. Instead, we only assume that the data point cloud can construct a graph which describes the manifold structure, and there are multiple concepts on different parts of the manifold. By maximizing the distance between different concepts and simultaneously preserving the local structure on the manifold, the learned metric can indeed give good ranking results. Moreover, we develop an online learning algorithm to incrementally learn the unlabeled data. Based on the relationship between of graph Laplacian and the Laplace-Beltrami operator on manifold, theoretical analysis of convergency is also presented.

Further author information: (Send correspondence to Bin Zhang and Yangqiu Song)

Bin Zhang: E-mail: zbin@cn.ibm.com, Telephone: +86 10 58748210
Yangqiu Song: E-mail: songyq99@mails.tsinghua.edu.cn, Telephone: +86 10 62796872
2. PROBLEM FORMULATION

We consider a semi-supervised setup of the distance metric learning problem. We denote the input points as \( D = \{X, Y\} \). There are \( l \) points have been appointed by human as the most certain points and \( u \) unlabeled points. Then, the observed input points can be written as \( X = (X_l, X_u) \), where \( X_l = (x_1, x_2, ..., x_l) \) and \( X_u = (x_{l+1}, x_{l+2}, ..., x_{l+u}) \). Each point \( x \in \mathbb{R}^d \) is a \( d \)-dimensional vector. The goal of our linear transformation algorithm is to find a projection matrix \( W \in \mathbb{R}^{d \times m} \) that transform the original Euclidean space to a more informative space for ranking.

We define \( G = (V, E) \) as a weighted neighborhood graph to describe point cloud \( X \). \( V \) is the vertex set of graph. \( E \) is the edge set which contains the pairs of neighboring vertices \((x_i, x_j)\). The neighboring vertices can be defined as: either \( \|x_i - x_j\| < r \) or \( x_j \) \((x_i)\) is among \( k \) nearest neighbors of \( x_i \) \((x_j)\). A typical adjacency matrix \( A \) of neighborhood graph is:

\[
A_{ij} = \begin{cases} 
\exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2}\right) & \text{if } (x_i, x_j) \in E \\
0 & \text{otherwise}
\end{cases}
\]  

(1)

Then the normalized graph Laplacian of a neighborhood graph is:

\[
L = I - D^{-1/2}AD^{-1/2},
\]

(2)

where the diagonal matrix \( D \) satisfies \( D_{ii} = d_i \), and \( d_i = \sum_{j=1}^{l+u} A_{ij} \) is the degree of vertex \( x_i \). Here the adjacency matrix and the normalized graph Laplacian are both symmetric.

3. METRIC LEARNING FOR RANKING

3.1 One Dimensional Projection

We first consider the one dimensional projection case. Based on the information provided by user and the observed data points, we want to find a projection that can balance two terms. The first term is the force that makes the points certainly “good” and certainly “bad” mostly separated. The second term is considered as a spring that preserves the intrinsic structure of the data point cloud.

To maximize the distances between the projected values of the most certainly “good” and “bad” points, we have the following objective function:

\[
\frac{l_1l_2}{l} (w^T m_1 - w^T m_2)^2 = l_1 (w^T m_1 - w^T m)^2 + l_2 (w^T m_2 - w^T m)^2 = w^T X_l L_l X_l^T w.
\]

(3)

where \( l_i \) is the number of data in class \( i \), \( w \) is the projection direction, \( m_1 \) is the mean of most certainly “good” points and \( m_2 \) is the mean of most certainly “bad” points. The graph Laplacian \( L_b \) is defined based on the graph which is constructed by the labeled data. Specifically it is \( L_b = D_b - A_b \), where

\[
(A_b)_{ij} = \begin{cases} 
\frac{1}{l} - \frac{1}{l_k} & x_i \text{ and } x_j \text{ belong to the same concept} \\
\frac{1}{l} & \text{otherwise}
\end{cases},
\]

(4)

and \( (D_b)_{ii} = \sum_j (A_b)_{ij} \). Actually, \( D_b \) is a zero matrix since the sum of rows of \( A_b \) are zeros.

The term (3) is equivalent to the between-class scatter of two-class LDA (Fisher discriminant analysis) in form. However, they have different meanings. In our algorithm, we want to separate the distance between the most certain points. In Fisher, its meaning is to separate the two classes in the projection direction. We do not assume there are classes on the manifold. Instead, we only want to find a direction, which is along the manifold structure, based on the most certain points.

Since we have defined the weighted graph (1) and (2), we formulate the following term to use the geometric information provided by both labeled and unlabeled data:

\[
w^T X L X^T w = \sum_{i,j} A_{ij} (\frac{1}{\sqrt{d_i}} w^T x_i - \frac{1}{\sqrt{d_j}} w^T x_j)^2
\]

(5)
Minimizing this term means that we desire projected points to be close if they are close in the input space, since we multiply a large weight \( A_{ij} \) if \( x_i \) and \( x_j \) are close. This term is also density-resisting because we multiply an inverse root of degree \( 1/\sqrt{d_i} \) to each projected points. When some region of the data is sparse, we give large weights to the points. Thus, the projection direction will pay more attention to this region to make the points in it be closer.

After the above analysis, we can give the criterion of the projection vector:

\[
 w^* = \arg \max_{w \in \mathbb{R}^d} \frac{w^T X_L L_b X_L^T w}{w^T X L X^T w}. \tag{6}
\]

Maximizing this objective function means that we want to both mostly separate the certain points and preserve the local structure of a manifold and the intrinsic geometry of the data point cloud. The projection direction is desired as along the manifold.

Like Fisher discriminant analysis, the solution is given by the eigenvector corresponding to the largest eigenvalue of \( (X L X^T)^{-1} X L L_b X^T w^* \). It is also the generalized eigenvector associated to the generalized eigenvalue of the following generalized eigenvalue problem:

\[
 X_L L_b X_L^T w^* = X L X^T w^*. \tag{7}
\]

This projection direction can be directly used for ranking the observed data as an order. The projected value on this direction is the rank value. For new test data, it is also possible to project them onto this direction to get their positions or retrieve the most similar points.

### 3.2 Multi-Dimensional Projection Problem

In many cases, instead of having the most certainly “good” and certainly “bad” points, we also have the most certainly labels that the points belong to one of the concepts. Here we use the name “concept” instead of “class” to distinguish the meaning of “ranking” and “classification”. A manifold may have multiple concepts and user can label most sure points of each concept. In this case, the concept number of the data is not two, and the projection vector number may also be not one.

We make use of the following criterion to find the subspace associate to the learned metric:

\[
 W^* = \arg \max_{W \in \mathbb{R}^{d \times m}} \frac{|W^T X_L L_b X_L^T W|}{|W^T X L X^T W|}. \tag{8}
\]

Note that the normalized graph Laplacian is independent to the “concept”, it only describes the local structure of the data manifold. Conversely, the graph constructed by labeled data based on (4) should be extended to multi-concept case. Then the matrix has the following property:

\[
 tr(W^T X_L L_b X_L^T W) = \sum_k w_k^T X_L L_b X_L^T w_k \\
= \sum_k \sum_{i,j} (A_{kj}) (w_k^T x_i - w_k^T x_j)^2 \\
= \sum_{j=1}^c l_j (m_j - m)^T W W^T (m_j - m) \\
= \sum_{j=1}^c l_j (m_j - m)^T M (m_j - m). \tag{9}
\]

\*Since the normalized graph Laplacian is semi-definite, we should add an extra regularization here. In practice, we have \( L = L + \delta I \).
The trace measures the variance of the projected mean vectors, which is also the variance under the new Mahalanobis distance metric. Maximizing this term is to maximize the distances among the most certain concepts.

Moreover, the term

$$\text{tr}(W^T XLX^T W) = \sum_k w_k^T XLX^T w_k$$

$$= \sum_k \sum_{i,j} A_{ij} \left( \frac{1}{\sqrt{d_i}} w_k^T x_i - \frac{1}{\sqrt{d_j}} w_k^T x_j \right)^2$$

$$= \sum_{i,j} A_{ij} \frac{1}{\sqrt{d_i}} z_i - \frac{1}{\sqrt{d_j}} z_j \|z\|^2$$

$$= \sum_{i,j} A_{ij} \left( \frac{1}{\sqrt{d_i}} x_i - \frac{1}{\sqrt{d_j}} x_j \right)^T M \left( \frac{1}{\sqrt{d_i}} x_i - \frac{1}{\sqrt{d_j}} x_j \right)$$

measures locally weighted variance, where the original high-dimensional data point $x$ is transformed into a low-dimensional vector $z = W^T x$. Minimizing this term means that the local structure of manifold under the new Mahalanobis distance metric consists with the manifold structure in the original Euclidean space.

The objective function (8) has the solution:

$$XLX^T w^*_j = \eta_j X_L L b X_L^T w^*_j, \quad j = 1, ..., m,$$

where $w^*_j (j = 1, ..., m)$ are the eigenvectors corresponding to the $m$ largest eigenvalues ($\eta_j$) of $(XLX^T)^{-1} X_L L b X_L^T$.

Having the new distance metric (or the low-dimensional vectors), we can choose several ranking algorithm to generate the rank value. For instance, we can use linear weighted combination of reduced features, or can use the non-linear regression method such as spline models or Gaussian processes. To handle the new query data, we can also use Gaussian process since it has the ability of induction. However, since the new distance metric itself contains the rank information, we can simply use the distances between the new points and observed data as the rank values.

### 3.3 Extension to Classification

It is straight forward that we can develop a semi-supervised metric learning algorithm for classification. The key problem is that we should consider about both how to use the classification information provided by the labeled data and how to add the information provided by the unlabeled data. One way is to seek the subspace that can both maximize the between-class scatter and minimize the within-class scatter, simultaneously it can also preserve the local structure of a manifold and the intrinsic geometry of the data point cloud. We use a parameter to balance the terms that can affect the final projection matrix:

$$W^* = \arg \max_{W \in \mathbb{R}^{d \times m}} \frac{|W^T S_b W|}{|W^T (S_w + \lambda XLX^T) W|}.$$ 

(12)

Then the optimal solution is given by:

$$(S_w + \lambda XLX^T) w^*_j = \eta_j S_b w^*_j \quad j = 1, ..., m,$$

where $w^*_j (j = 1, ..., m)$ are the eigenvectors corresponding to the $m$ largest eigenvalues of $(S_w + \lambda XLX^T)^{-1} S_b$.

This method is closely related to the regularized discriminant analysis model.
4. ONLINE LEARNING OF UNLABELED DATA

In this section, we present an online learning algorithm which can deal with the incrementally changed unlabeled set. To make the algorithm practicable, we make some modifications:

1. We adopt the following objective function:

\[ J(W) = tr(W^T X_L L b X_U^T W - \frac{1}{(l + u)^2} W^T X L X^T W). \]  

By taking the derivative, the gradient is given by:

\[ \frac{\partial J(W)}{\partial W} = 2(X_L L b X_U^T - \frac{1}{(l + u)^2} X L X^T)W. \]

2. We utilize graph Laplacian \( L = D - A \) instead of the normalized one. And we make use of a full connected graph and the corresponding adjacency is \( A_{ij} = \exp\{-\frac{||x_i - x_j||^2}{2\sigma^2}\} \) for all pairwise points in the observed data set. It has been proven that both normalized and un-normalized graph Laplacain can approximate the Laplace-Beltrami operator on the manifold in some conditions (Lemma 1, 13, 14). Then we have:

\[
X(D - A)X^T = \sum_i D_{ii}x_i x_i^T - \sum_{ij} A_{ij} x_i x_j^T
\]

\[
= \sum_{ij} A_{ij} (x_i x_i^T - x_i x_j^T)
\]

\[
= \sum_{ij} A_{ij} (x_i x_i^T - 2x_i x_j^T + x_j x_j^T)
\]

\[
= \sum_{ij} A_{ij} (x_i - x_j)(x_i - x_j)^T. \]  

3. Instead of taking the derivative of (14) as zeros, we make use of the recursive rule:

\[ W(t + 1) = W(t) + \mu \frac{\partial J(W)}{\partial W}(t), \]  

where \( \mu \) is the step size. In each iteration \( t + 1 \), we randomly select an unlabeled data \( x(t + 1) \). Then, we denote the data set as \( X_U(t + 1) = (X_U(t), x(t + 1)) \), and \( X(t + 1) = (X_L, X_U(t + 1)) \). The number of the data points in iteration \( t + 1 \) is \( l + u(t + 1) \). Therefore, we have:

\[
X(t + 1)L(t + 1)X^T(t + 1)
\]

\[
= X(t)L(t)X^T(t) + 2 \sum_{i=1}^{l+u(t)} A_{i,l+u(t)+1}(x_i - x(t + 1))(x_i - x(t + 1))^T. \]

Based on the analysis and approximations above, we have the online algorithm to deal with the unlabeled data. The flowchart is shown in Table 1. Moreover, the following Lemmas and Theorem guarantee the convergence of our algorithm.

**Lemma 1.** (Taken from Theorem 1.) Let data points \( (x_1, x_2, ..., x_n) \) be sampled from a uniform distribution on a manifold \( M \subset \mathbb{R}^d \). Put \( n = n^{-\frac{k-1}{d}} \), where \( \alpha > 0 \) and \( k \) is the inherent dimensionality of \( M \). Let \( f \in C^\infty(M) \). Then there is a constant \( C \), s.t. in probability,

\[
\lim_{n \to \infty} C (\frac{4\pi\sigma_n}{n})^{\frac{k+2}{d}} L(n, \sigma_n) f(x) = \Delta_M f(x)
\]  

SPIE-IS&T Vol. 6822  682225-5
1. Input: \( X(0) = (X_L, X_U(0)), Y = Y_L \).
2. Initialize \( W(0) \) as the eigenvectors corresponding to the first largest eigenvalues of \( (X(0)L(0)X(0)^T)^{-1}X_LL_bX_U^T \).
3. In iteration \( t + 1 \):
   a) Read a point \( x(t + 1) \), and calculate the new elements of the adjacency matrix:
   \[ A_{i,l+u(t+1)} = \exp\left\{ -\frac{||x_i - x(t+1)||^2}{2\sigma^2} \right\}. \]
   b) Let \( u(t+1) = u(t) + 1 \), \( X_U(t+1) = (X_U(t), x(t+1)) \), and \( X(t+1) = (X_L, X_U(t+1)) \). Calculate the new graph Laplacian term:
   \[ v(t+1) = x_i - x(t+1) \quad \text{and} \quad B(t+1) = v(t+1)v(t+1)^T \]
   and
   \[ \frac{1}{(t+u(t))^2}X(t+1)L(t+1)X(t+1)^T \]
   \[ = \frac{1}{(t+u(t))^2}X(t)L(t)X(t)^T + \frac{1}{(t+u(t))^2}2\sum_{i=1}^{t+u(t)}A_{i,l+u(t+1)}B(t+1). \]
   c) Find a new solution based on the instant gradient:
   \[ W(t+1) = W(t) + \mu X_L L_b X_U^T W(t) - \frac{\mu}{(t+u(t))^2}X(t+1)L(t+1)X(t+1)^T W(t). \]
   d) Normalize \( W(t+1) \) as \( W(t+1)^T W(t+1) = I \).
   e) If the Frobenius norm \( ||W(t+1) - W(t)||_F \geq \varepsilon, i = t + 2 \). \( \varepsilon \) is a stop parameter.
4. Output: \( W_{\text{Final}} \).

Table 1. Flowchart of online learning algorithm with unlabeled data.

where \( L(n) \) depends on \( \lambda_n \) since \( A_{ij}(n) \) is defined as \( A_{ij}(n) = \exp\left\{ -\frac{||x_i - x_j||^2}{2\sigma^2} \right\} \) and \( \Delta_M \) is the Laplace-Beltrami operator on the manifold.

The proof is given in.\(^{13} \)

**Lemma 2.** Let \( W = (w_1, w_2, ..., w_m) \) and restrict the function to be linear \( f_i(x) = w_i^T x \) and \( f_i = X^T w_i \). The smoothness of a manifold in the projected space can be measured using:

\[
\lim_{n \to \infty} \frac{C}{n^2} w_i^T X(n)L(n)X(n)^T w_i = \int_{\mathcal{M}} f_i^T \Delta_\mathcal{M} f_i
\]

where \( C \) is a constant.

**Proof:** Since the the Laplace-Beltrami operator can be approximated by the graph Laplacian due to **Lemma 1, Lemma 2** is natural to generalize the discrete case to the continuous case. Moreover, it is the regularization term:\(^{14} \)

\[
\int_{\mathcal{M}} ||\nabla_\mathcal{M} f_i|| = \int_{\mathcal{M}} \langle \Delta_\mathcal{M} f_i, f_i \rangle = \int_{\mathcal{M}} f_i^T \Delta_\mathcal{M} f_i. \quad (21)
\]

**Theorem 1.** \( W(t) \) being the eigenvectors of \( (\int_{\mathcal{M}} X(\infty) \Delta_\mathcal{M} X(\infty)^T)^{-1}X_L L_b X_U^T \) is the fixed point of equation (14).

**Proof:** Due to **Lemma 2** we have:

\[
\lim_{n \to \infty} \frac{C}{n^2} (W^T X(n)L(n)X(n)^T W) = W^T (\int_{\mathcal{M}} X(\infty) \Delta_\mathcal{M} X(\infty)^T) W. \quad (22)
\]
By using the following update function

$$W(t + 1) = W(t) + \mu \left( X_L L_b X_T^T - \frac{1}{n(t + 1)^2} X(t + 1)L(t + 1)X^T(t + 1) \right) W(t)$$

(23)

we have

$$W(t + 1) = W(t)$$

(24)

if \( W(t) \) are the eigenvectors of \( (\int_M X(\infty)\Delta_M X(\infty)^T)^{-1}X_L L_b X_T^T \) when \( t \to \infty \). Note that \( W(t + 1) \) should be normalized by Gram-Schmidt orthogonalization which can be implemented by QR factorization in each iteration.

\[\blacksquare\]

5. EXPERIMENTS ON BENCHMARKS

5.1 Retrieval Accuracy

We test the algorithms with the digits image retrieval problem. User submit a query example and the computer retrieves and ranks the points in the observed data base. The retrieval accuracy is defined as:

$$\text{Accuracy} = \frac{\text{relevant examples in top } N \text{ returns}}{N}.$$ 

(25)

5.2 USPS Data Set

We utilize the USPS handwritten digits images\(^\dagger\) as the benchmark to test the efficiency of our algorithms. The original database contains 7291 training data and 2007 test data, and each data point is an image with 16 × 16 resolution. All data are “0”-“9” digit representations.

5.3 Comparison

We first test our algorithms with PCA, LDA and LPP. We randomly select 3000 data points as seen and 3000 data points unseen. Then seen data is randomly split into labeled and unlabeled data. For unsupervised methods PCA and LPP, we use the seen data to find the projection vectors. For supervised method LDA, we use the labeled data in the first \( d - 1 \) dimensions in PCA subspace as input features. For our semi-supervised distance metric learning algorithm (SSDML), we make use of the partially labeled seen data in the \( d - 1 \) dimensions in

\(^\dagger\)http://www.kernel-machines.org
PCA sub-space as input features. New distance metric is adopted to test the retrieval accuracy on the unseen set. The retrieved number is set to 20. Results are shown in Fig. 1 (a). Each test accuracy is an average of 50 random trials. We see that SSDML is competitive with PCA and LPP, and significantly better than LDA. We also vary the retrieved numbers to show the accuracy differences. We select retrieved numbers as 5, 20, 50 and 100 and the average accuracy rates of 50 random trials are shown in Fig. 1 (b). As expectation, the accuracy decreases when retrieved number increasing.

5.4 Online Learning with Unlabeled Data
To test the online learning algorithm, we randomly select 8000 points as the seen data, and use the remains as the test unseen data. The retrieved number is set to 20. Fig. 2 (a) and (b) show that accuracies vary with different steps and different labeled numbers. The retrieval accuracy rates can converge to fixed values. Note that the algorithm only gets to local maximum since it is a gradient based method.

6. CONCLUSIONS
In this paper, we present a novel dimensionality reduction algorithm for ranking, where only most certain points are provided by user. The essence of the proposed algorithm is a semi-supervised learning method. However, it is different from the start point of traditional semi-supervised classification and clustering since we do not make the assumption of existence of classes and do not want to find the classification boundaries. Our methods can perform outstanding results in the digits image retrieval problem.

REFERENCES