# A Transductive Framework of Distance Metric Learning by Spectral Dimensionality Reduction

**Fuxin Li** FUXIN.LI@IA.AC.CN
Laboratory of CSIS, Institute of Automation, Chinese Academy of Sciences, Beijing, China

**Jian Yang** JIANYANG@BJUT.EDU.CN
International WIC Institute, Beijing University of Technology, Beijing, China

**Jue Wang** JUE.WANG@IA.AC.CN
Laboratory of CSIS, Institute of Automation, Chinese Academy of Sciences, Beijing, China

## Abstract

Distance metric learning and nonlinear dimensionality reduction are two interesting and active topics in recent years. However, the connection between them is not thoroughly studied yet. In this paper, a transductive framework of distance metric learning is proposed and its close connection with many nonlinear spectral dimensionality reduction methods is elaborated. Furthermore, we prove a representer theorem for our framework, linking it with function estimation in an RKHS, and making it possible for generalization to unseen test samples. In our framework, it suffices to solve a sparse eigenvalue problem, thus datasets with $10^5$ samples can be handled. Finally, experiment results on synthetic data, several UCI databases and the MNIST handwritten digit database are shown.

## 1. Introduction

The problem of distance metric learning has gained considerable interest in recent years (Xing et al., 2003; Zhang, 2003; Kwok & Tsang, 2003; Wu et al., 2005; Weinberger et al., 2006; Zhang et al., 2006; Hoi et al., 2006; Bach & Jordan, 2006; Lebanon, 2006). Distance metric learning seeks to improve an apriori metric (often the Euclidean metric) by adapting it to fit a certain training set. In this way, it is possible to discriminate relevant and irrelevant features and the performance of distance-based methods, such as $k$-nearest-neighbors (kNN) or spectral clustering, can be improved.

The problem of blending apriori knowledge and knowledge from training data has been haunting pattern recognition for many years and the paradigm of distance metric learning provides a potential way to solve it. In this paradigm, a new metric is learned from the prior metric and standard methods can be applied easily on the learned metric. This learned metric may again be used as a prior metric and the learning process would continue endlessly. This prospect urges us to study this problem in depth.

Many current distance metric learning methods seek to learn a Mahalanobis metric in the input space (Xing et al., 2003) or the feature space (Kwok & Tsang, 2003; Wu et al., 2005; Hoi et al., 2006). However, to get a Mahalanobis metric is equivalent to linearly transform the input data and take the Euclidean metric in the transformed space. This draws our interest in comparing them with dimensionality reduction methods. Dimensionality reduction methods do a linear or nonlinear transformation on the input to get a low-dimensional representation of the data. A distance metric in the transformed space can readily be seen as a metric *learned* by the dimensionality reduction algorithm. In fact, one dimensionality reduction algorithm, Laplacian Eigenmaps (Belkin & Niyogi, 2003), solved an optimization problem with a goal very similar to distance metric learning.

However, many dimensionality reduction methods were unsupervised, which means they did not take the label information of the training data into account. Can distance metric learning be seen as a supervised or semi-supervised improvement of these unsupervised methods? This is what we try to answer in our work.

Appearing in *Proceedings of the $24^{th}$ International Conference on Machine Learning*, Corvallis, OR, 2007. Copyright 2007 by the author(s)/owner(s).

In this paper, we propose a transductive framework of distance metric learning. Under certain assumptions, it can be directly connected with many spectral dimensionality reduction methods, such as kernel PCA, MDS, ISOMAP (Tenenbaum et al., 2000) and Laplacian Eigenmaps. Our method only need to solve a sparse eigenvalue problem, thus it scales much better than many current distance metric learning and semi-supervised learning methods.

Furthermore, we proved a representer theorem for our framework, thus connecting it to function estimation in a reproducing kernel Hilbert space (RKHS). An interesting point is that the loss function in our representer theorem is a squared loss penalizing pairwise differences, which is different from almost all previous regularization methods.

We will present our framework and show these connections in Section 2. In Section 3 we will discuss the design of cost and penalty functions in the framework. Related works will be summarized in Section 4, and experiments will be shown in Section 5. Discussions and concluding remarks will be given in Section 6.

## 2. A Transductive Framework of Distance Metric Learning

We first introduce some notation conventions. Matrices and functions are represented by upper case letters, column vectors by lower case letters. The vector of all ones, the identity matrix, and the matrix of all ones are denoted by $e$, $I$ and $E$ respectively. $A^+$ denotes the Moore-Penrose pseudo-inverse of a matrix $A$. And $diag(x)$ is a diagonal matrix whose diagonal is $x$.

### 2.1. The Framework

Given $n$ labeled samples $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ and $m$ unlabeled samples $x_{n+1}, \ldots, x_{n+m}$, we can formulate the transductive distance metric learning problem as an optimization problem:

$$\min_D \frac{1}{n} \sum_{i=1}^{n} C(D, x_i, y_i) + \lambda \Omega(D), \qquad (1)$$

where $C(D, x_i, y_i)$ is an arbitrary cost function concerning $D$ and labeled item $(x_i, y_i)$, $\Omega(\cdot)$ is a penalty function based on apriori knowledge of the metric, and $\lambda$ is a parameter controlling the strength of the penalty. Intuitively, we try to balance prior knowledge and knowledge from item labels, by which wrong prior beliefs can be corrected by reliable observations.

Since there are only $n + m$ samples on which we can measure the cost and penalty functions, the target distance metric $D$ can be reduced to an $(n+m) \times (n+m)$ distance matrix with entries $d_{ij} = d^2(x_i, x_j)$. We denote this matrix as $D$ too.

In this work we use a special squared cost function by letting $C(D, x_i, y_i)$ and $\Omega(D)$ linear with $d^2$. Setting $C(D, x_i, y_i) = \sum_{j=1}^{n} \sum_{k=1}^{n} c_{ijk} d_{jk}$ and $\Omega(D) = \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij} d_{ij}$, we can collect coefficients $c_{ijk}$ and $p_{ij}$ into symmetric matrices $C_i$ and $P$, and rewrite the optimization problem (1) in matrix form:

$$\min_D \operatorname{Tr}(\frac{1}{n} \sum_{i=1}^{n} C_i D + \lambda P D). \qquad (2)$$

The details of the cost and penalty functions used in this paper are deferred to section 3 since they do not have much to do with the framework.

To solve the problem efficiently, we have to impose some structures on $D$. The assumption in this paper is the *Euclidean assumption*, which says that the matrix $D$ must be Euclidean, defined as follows:

**Definition 1** *(Gower & Legendre, 1986; Zhang, 2003) An $n \times n$ matrix $D$ is Euclidean if there are $n$ points $x_i (i = 1, \cdots, n)$ which can be embedded in a Euclidean space where the squared Euclidean distance between $x_i$ and $x_j$ is $d_{ij}$.*

The distance matrix of any finite set of points in a Euclidean space is Euclidean. Recall that every Euclidean distance metric can be induced by an inner product. That is, $d^2(x, y) = \langle x - y, x - y \rangle = \langle x, x \rangle + \langle y, y \rangle - 2\langle x, y \rangle$ for some real inner product $\langle \cdot, \cdot \rangle$. Similarly, every Euclidean distance matrix can be induced by a Gram matrix $G$, with $g_{ij} = \langle x_i, x_j \rangle$, as the following proposition shows:

**Proposition 2** *(Gower & Legendre, 1986; Zhang, 2003) A matrix $D$ is Euclidean if and only if its associated Gram matrix $G = -\frac{1}{2}(I - \frac{1}{n}E) D (I - \frac{1}{n}E)$ is positive semi-definite.*

The proof of this proposition involves only simple algebra to verify that $D$ is indeed the distance matrix induced by $G$. Namely, $d_{ij} = (g_{ii} + g_{jj}) - 2g_{ij}$. In this way, learning an Euclidean distance metric is linked to the well-studied problem of learning a kernel from data (e.g., (Lanckriet et al., 2004)). However, the cost function and penalty terms are different from previous works that tried to learn an optimal kernel for SVM.

By Proposition 2, we can turn (2) into an optimization problem on the Gram matrix $G$. That is to say, we need to find matrices $C_i'$ and $P'$ that satisfies $\operatorname{Tr}(C_i' G) = \operatorname{Tr}(C_i D)$ and $\operatorname{Tr}(P' G) = \operatorname{Tr}(P D)$. The rule of constructing $C_i'$ and $P'$ is given by the following proposition:

**Proposition 3** *For symmetric matrix $A$, Euclidean matrix $D$ and its corresponding Gram matrix $G$, we have* $\mathrm{Tr}(AD) = \mathrm{Tr}(A'G)$ *if* $A' = 2(diag(Ae) - A)$.

**Proof** It is easy to see that $A'E = 0$ since $A'$ has zero row sums. Therefore

$$\begin{aligned}\mathrm{Tr}(A'G) &= \mathrm{Tr}(-\frac{1}{2}A'(I-\frac{1}{n}E)D(I-\frac{1}{n}E)) \\ &= \mathrm{Tr}(-\frac{1}{2}(I-\frac{1}{n}E)A'(I-\frac{1}{n}E)D) \\ &= \mathrm{Tr}((A-diag(Ae))D) = \mathrm{Tr}(AD).\end{aligned}$$

The last equation follows from the fact that the diagonal of $D$ is zero since it corresponds to the distance from an item to itself. □

After constructing $C_i'$ and $P'$, we can rewrite (2) into an optimization problem on the Gram matrix $G$. Since $G$ is positive semi-definite, it has an orthogonal decomposition $G = XX^T$, with the additional normalization constraint $\mathrm{Tr}(G) = \mathrm{Tr}(X^TX) = 1$, we get the following eigenvalue problem:

$$\begin{array}{ll}\min\limits_{X} & \mathrm{Tr}(X^T(\sum_{i=1}^n C_i' + \lambda P')X) \\ s.t. & X^TEX = 0 \\ & \mathrm{Tr}(X^TX) = 1.\end{array} \qquad (3)$$

The optimal solution of $X$ is then given by the eigenvectors corresponding to the smallest eigenvalues of $\sum_{i=1}^n C_i' + \lambda P'$. In fact, the $i$-th row of $X$ corresponds to the projection of $x_i$ into a low-dimensional Euclidean space and the learned distance metric is simply the Euclidean metric in this transformed space. The method is thus linked with spectral dimensionality reduction methods. This connection will be investigated further in the following subsections.

We summarize our framework in Algorithm 1. Different from many kernel methods, the smallest eigenvalues in our framework can be positive or negative without affecting the solution. In this case the correct implementation is to discard the all-one eigenvector corresponding to eigenvalue 0 and take other eigenvectors.

### 2.2. Relations with other Spectral Methods

The link of our framework to Laplacian Eigenmaps is the most obvious one. One only need to set $C_i' = 0$ and $P' = L$, the graph Laplacian, to get exactly the Laplacian Eigenmap method. Since Laplacian Eigenmaps can be considered as Kernel PCA ((Ham et al., 2004; Bengio et al., 2004)) with $L^+$ as the kernel, the relationship of our framework and Kernel PCA becomes clear: set $C_i' = 0$, $P' = K^+$ and we get Kernel PCA on centralized kernel $K$. In a similar way, other spectral dimensionality reduction methods such as ISOMAP and LLE can be brought into our framework.

**Algorithm 1** The Framework

```
Design the cost matrices $C_i$ and penalty matrix $P$.
Construct $C_i'$ and $P'$ by Proposition 3.
Computer the eigenvectors of $\sum_{i=1}^n C_i' + \lambda P'$ and take $d$ eigenvectors corresponding to the smallest eigenvalues of the matrix, excluding the all-one eigenvector corresponding to eigenvalue 0. Denote the matrix with each column an eigenvector by $X$.
The $i$-th row of X are the coordinates in a $d$-dimensional Euclidean space of the $i$-th item.
```

Note that the penalty matrix $P'$ can often be considerably sparser than the kernel matrix. Hence it is much easier to compute eigenvectors of it. By using Krylov space methods, our framework is able to handle datasets with $10^5$ samples, thus making it much more scalable than almost all semi-supervised learning methods.

### 2.3. A Representer Theorem

Suppose that $\mathcal{H}$ and $\tilde{\mathcal{H}}$ are RKHSs with inner products $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ and $\langle\cdot,\cdot\rangle_{\tilde{\mathcal{H}}} = \langle\cdot,\cdot\rangle_{\mathcal{H}} + S(\cdot)^TMS(\cdot)$, where $S : \mathcal{H} \to \mathbb{R}^{n+m}$ is the evaluation map on the labeled and unlabeled data, $S(f) = (f(x_1), f(x_2), \ldots, f(x_{n+m}))$, and $M$ a positive semi-definite penalty matrix. Let $k(\cdot,\cdot)$ be the representer of $\mathcal{H}$ and $K$ be the $(n+m) \times (n+m)$ Gram matrix, $k_{ij} = k(x_i, x_j)$.

To optimize a cost function based on pairwise distances for one-dimensional embeddings, we can have the following regularization problem:

$$\begin{array}{ll}\min\limits_{f\in\tilde{\mathcal{H}}} & \sum_{i,j=1}^n w_{ij}(f(x_i) - f(x_j))^2 + \lambda\|f\|_{\tilde{\mathcal{H}}}^2 \\ s.t. & S(f)^TS(f) = 1\end{array} \qquad (4)$$

where $w_{ij}$ are weights generated from label information controlling the cost on $x_i$ and $x_j$. The next theorem is the main theoretical result of this paper.

**Theorem 4** *Suppose that the Gram matrix $K$ of rank $n+m-1$ satisfies $Ke = 0$. Let the penalty matrix and cost matrices in (3) satisfy $P' = K^+ + (I - \frac{1}{n}E)M(I - \frac{1}{n}E)$ and $\sum_{i=1}^n C_i' = \sum_{i,j=1}^n w_{ij}(s_i - s_j)(s_i - s_j)^T$, where $s_i$ is a vector with $1$ in the $i$-th position and $0$ in other positions. If the one-dimensional optimal solution of (3) is $x^*$ and the optimal solution of (4) is $f^*$, then we have $x^* = S(f^*)$.*

The proof of the theorem is given in the appendix.

This theorem relates our framework with function estimation in an RKHS. From this theorem, it is possible to give a natural out-of-sample extension of our algorithm by RKHS techniques (See Lemma 5 and Lemma 6 in the Appendix). Note that the case of multi-dimensional embeddings can follow from the representer theorem for vector-valued functions in (Micchelli & Pontil, 2005).

A significant difference from previous regularization methods is that we do not try to specify the loss of $f(x)$ to some classification or regularization goal $y$. Instead, we penalize pairwise differences. In multiclass classification, one cannot reasonably specify a $y \in \mathbb{R}$, therefore the regularization problem (4) seems more sensible than previous approaches (e.g., (Zhou et al., 2004)).

## 3. Design of the Cost and Penalty Functions

### 3.1. The Cost Function

Intuitively, for distance-based methods, distances between items of the same class should be smaller than distances between items of different classes. This also resembles the idea in discriminant analysis.

Consider labeled data $\{(x_i, y_i)\}_{i=1,\dots,n}$. Following the intuition, we define a cost function $C$ between two distances $d_{ij}$ and $d_{ik}$ as

$$C(d_{ij}, d_{ik}) = \begin{cases} d_{ij} - d_{ik}, & y_i = y_j \text{ and } y_i \neq y_k \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

Note that $C(d_{ij}, d_{ik})$ can be negative. This problem can be resolved by specifying a minimum and a maximum value of each $d_{ij}$ and adding corresponding constants to the cost function. However we have not done this to simplify notations.

To preserve intra-class structures, we only evaluate the cost function between neighboring items. For each item $(x_i, y_i)$, we take $k$ intra-class neighbors and $k$ inter-class neighbors of it. By averaging (5) over these neighbors, we get the cost function

$$C(D, x_i, y_i) = \frac{1}{k} \sum_{x_j \in \mathcal{N}_k^{in}(x_i)} d_{ij} - \frac{1}{k} \sum_{x_j \in \mathcal{N}_k^{out}(x_i)} d_{ik},$$

where $\mathcal{N}_k^{in}(x_i)$ is the set of $k$ intra-class neighbors of $x_i$ and $\mathcal{N}_k^{out}(x_i)$ is the set of $k$ inter-class neighbors of $x_i$.

To simplify notations, we put the matrix $\sum_{i=1}^n C_i$ into a single symmetric matrix $C$ with entries

$$c_{ij} = \begin{cases} \frac{1}{k}, x_j \in \mathcal{N}_k^{in}(x_i) \\ -\frac{1}{k}, x_j \in \mathcal{N}_k^{out}(x_i). \end{cases} \quad (6)$$

It can be verified that $\mathrm{Tr}(CD) = \sum_{i=1}^n C(D, x_i)$ and the cost matrix also satisfies the corresponding conditions in Theorem 4.

If there are very few labeled items in the training set, one may not be able to find $k$ intra-class or inter-class neighbors for some $x_i$ and $k$. In this case, (5) is averaged only on the available neighbors and the entries in the cost matrix $C$ should be changed respectively.

### 3.2. The Penalty Function

From Theorem 4, we know that the penalty matrix should adopt the form $P = K^+ + (I - \frac{1}{n}E)M(I - \frac{1}{n}E)$, for some positive semi-definite matrix $M$ and Gram matrix $K$. The simplest choice would be setting $K^+ = M = L$, the Laplacian matrix. However, that would make it hard for the algorithm to generalize to new test points. Alternatively, one can use a general kernel such as the Gaussian kernel to compute $K$ and its pseudo-inverse $K^+$. However for datasets with $10^4$ samples or more, the computation of $K^+$ would be prohibitive on current PC due to memory constraints. We face a dilemma here: either to get the ability to generalize to new test points, or to get the ability to handle large scale datasets. We choose the latter here and used $P = L$ as the penalty matrix[1].

Using $L$ as the penalty matrix has an intuitive meaning, as pointed out by Belkin and Niyogi (2003). That is to punish large distances between points that should be nearby due to apriori knowledge. This procedure can be seen as some sort of geometric structure we want to preserve. Thus the framework can be seen intuitively as the trade-off between a prior geometric structure and a structure which comes from the training labels.

## 4. Related Works

The idea of balancing prior metric and label information is not new in distance metric learning(Zhang, 2003; Kwok & Tsang, 2003; Wu et al., 2005; Weinberger et al., 2006). The criterion used in our framework is very similar to those in (Zhang, 2003), (Kwok & Tsang, 2003) and (Wu et al., 2005). However their algorithms were slower and more sophiscated. Zhang (2003) used a metric MDS, while Kwok and Tsang

[1] If one want to use a general kernel, then the equivalent eigenvalue problem (7) in the Appendix might be a better choice since it involves fewer computation.

(2003) and Wu et al. (2005) used quadratic programming. Weinberger et al. (2006) proposed a hinge loss function and used semi-definite programming. However, these methods were all inductive methods that did not take advantage of unlabeled data.

Zhang et al. (2006) proposed two algorithms, GWPC and KTDA, that took the transductive setting. In GWPC, a Wishart process prior was assumed and parameters were estimated by an EM algorithm. The KTDA algorithm used similar assumptions and EM algorithm but had a different goal. Our criterion is similar to the one used in KTDA, but the assumptions and optimization method are different.

Besides that, Hoi et al. (2006) proposed Discriminative Component Analysis to learn a Mahalanobis matrix in the feature space and solved a generalized eigenvalue problem similar with kernel discriminant analysis. Our approach is different from theirs since we only solve a regular eigenvalue problem.

Our framework can also be seen as learning an optimal kernel from label information, which has been a popular topic in recent years. However, most methods seek to learn a kernel to optimize the performance of SVM, such as (Lanckriet et al., 2004). The kernel-target alignment method in (Cristianini et al., 2002) was the first to came up with the idea of "aligning" between the prior kernel and an ideal kernel that is given by label information. Bousquet and Hermann (2003) discussed the asymptotic behavior of learning kernel matrices, some of the bounds there are also applicable to our method.

## 5. Experiments

In this section, we will conduct experiments on synthetic and real data to test our algorithm, which we denote as TDL for Transductive Distance Learning. In the experiments, the regularization parameter $\lambda$ is determined by grid search, while other parameters are hand-set.

### 5.1. Synthetic Data

We use a variant of the now famous two moons dataset (Zhou et al., 2004) to test our method and Laplacian Eigenmaps. The original two moons are very easy for nonlinear spectral methods such as Laplacian Eigenmaps, so we use a harder version (Figure 1). The two moons are pushed nearer to each other so it is much more difficult to discriminant between them.

In our experiment, Laplacian Eigenmaps and TDL are tested on a two moons dataset with 200 samples. The normalized Laplacian of an adjacency graph is used for penalty, which is built with a Gaussian kernel $p_{ij} = \exp(-\frac{\|x_i - x_j\|^2}{0.25})$ on Euclidean distances. The parameter $\lambda$ in TDL is set to 16. We projected the data to one dimension for classification, however we draw both dimensions for the ease of visualization.

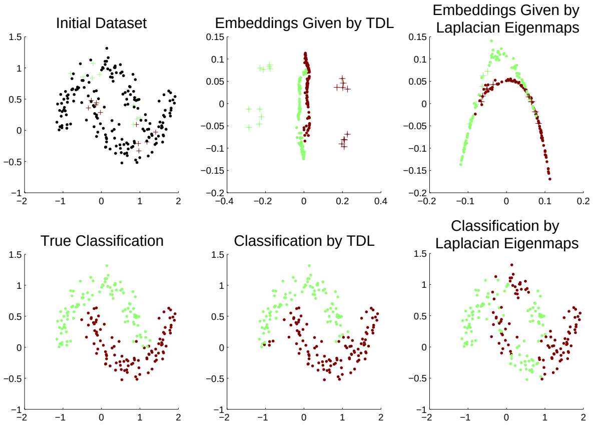

*Figure 1.* Experiment results on two moons. The classification is given by 1-NN on the first feature (horizontal axis).

From Figure 1, we can see that Laplacian Eigenmaps totally failed to discriminate the moons while TDL successfully discriminated them. An interesting point is that labeled samples were pushed far away from the unlabeled samples in TDL. This might be counterintuitive at first glance, but in fact it reflected a common sense in regularization that labeled data from different classes should be separated by large margins. Since the cost functions in TDL are only measured on the labeled samples, the unsmooth projection that pushed the labeled data away is inevitable. However, the result is good enough for classification, since the labeled data "pulled" the nearby unlabeled samples toward them by regularization, which actually separated the two classes. A more robust algorithm would need changing the cost function and will be pursued in future works.

### 5.2. UCI Databases

We experimented TDL and Laplacian Eigenmaps on four UCI databases: `Wisconsin breast cancer`, `Ionosphere`, `Sonar` and `Wine`. The input features are renormalized to range $[0, 1]$. Results are averaged over 100 random splits of the data with 10% for training and 90% for testing. The classification in the algorithms is given by a 1-NN rule. The construction of the

*Table 1.* Test set accuracies on UCI databases, the highest accuracies are shown in boldface.

| Method | Breast cancer | Ionosphere | Sonar | Wine |
|---|---|---|---|---|
| TDL | 94.74(±1.73) | **89.37**(±5.32) | 63.65(±5.68) | 93.09(±4.00) |
| Laplacian Eigenmaps | **94.80**(±1.61) | 85.01(±4.02) | 62.88(±4.91) | 93.48(±2.81) |
| GWPC | 94.58(±1.42) | 85.58(±5.63) | **70.45**(±4.73) | 93.79(±2.14) |
| KTDA | 94.47(±1.47) | 87.56(±5.73) | 70.22(±4.59) | **94.59**(±2.00) |
| KFDA | 93.30(±1.82) | 77.06(±10.12) | 67.07(±5.55) | 85.37(±7.83) |
| SVM | 93.35(±2.05) | 77.37(±10.00) | 67.07(±5.50) | 92.59(±3.29) |
| KNM | 90.89(±1.51) | 76.99(±7.78) | 65.63(±5.81) | 87.22(±5.32) |
| 1-NN | 92.94(±1.59) | 81.14(±4.27) | 69.18(±4.60) | 91.71(±3.00) |

cost and penalty matrices is the same as the synthetic data. The regularization parameter $\lambda$ is set to 1024 on all four datasets. For all datasets except `Breast cancer`, we take 10 eigenvectors. `Breast cancer` has only 9 features so we take 5 eigenvectors on it.

The results are shown in Table 1. The results for GWPC, KTDA, KFDA, SVM, KNM and 1-NN are taken from (Zhang et al., 2006). In which, GWPC and KTDA are the two algorithms proposed in (Zhang et al., 2006), KFDA is Kernel discriminant analysis, and KNM is kernel nearest mean classifier.

It can be seen that the results of TDL outperforms Laplacian Eigenmaps in some cases and are comparable with GWPC and KTDA. These results validated that in our method, information from labeled data are used properly to improve the distance metric. Moreover, TDL scales better than GWPC and KTDA. The result may get even better when more advanced distance-based methods, such as spectral clustering, are applied on the learned metric.

### 5.3. Incorporating Prior Knowledge

As we have discussed in the introduction, it is very important in distance metric learning to incorporate prior knowledge. However, a simple Euclidean metric is unsuitable to be the prior metric in many cases, and metrics that were designed from apriori knowledge of the underlying problems are more preferable. An excellent example of these metrics is Tangent Distance, which was designed for handwritten digit recognition (Simard et al., 1998). In that task there are many transformations that are irrelevant to the classification of digits, such as translation, scaling, rotation and thickening. In tangent distance, these transformations were approximated by tangent vectors of a parameterized manifold, and the metric were made invariant to them. Albeit its simplicity, tangent distance held the record on handwritten digit databases for many years.

*Table 2.* Test set accuracies on the MNIST database. The two columns show the results with Euclidean distance and Tangent distance as apriori information.

| Method | Euclidean | Tangent |
|---|---|---|
| TDL | **96.13**(±0.07) | **97.49**(±0.04) |
| Laplacian Eigenmaps | 95.91(±0.07) | 97.41(±0.03) |

The tangent distance is a metric for unsupervised data, which made it ideal to serve as prior knowledge in our algorithm. In this subsection, we do experiments on the MNIST database trying to improve tangent distance using our framework. The MNIST database (LeCun et al., 1998) consists 60000 training samples and 10000 test samples. In our experiments we use all the 70000 samples and take a random 5% as labeled. The results are averaged over 10 random splits.

We use the cost matrix (6) with $k = 20$. The penalty function is the Laplacian constructed from 20 nearest-neighbors. 60 eigenvectors are taken with $\lambda = 128$. The classification is given by a 10-NN rule. For computing the tangent distance, the hierarchical method (Simard et al., 1998) is used to speed-up the computation.

Since the size of the MNIST database is very large, few semi-supervised methods can be ran on it. We compare TDL with Laplacian Eigenmaps, which is known to have an excellent semi-supervised performance on this database (Belkin & Niyogi, 2003). The results are shown in Table 2, it can be seen that TDL showed a little improvement over Laplacian Eigenmaps. The improvement is smaller when tangent distance is used, since the distance metric already discriminated the samples very well. Furthermore, the time used in the computation is acceptable. Given the Laplacian matrix as input, the computation time is about 8 minutes for a single run on a 2.8 GHz Pentium IV PC.

## 6. Discussion and Conclusion

In this paper, we elaborated on the connection between distance metric learning and spectral dimensionality reduction. We showed that under the Euclidean assumption the problem of distance metric learning can be solved by spectral dimensionality reduction methods with label information injected. This opens up possibilities in both directions: One can now consider the case where the Euclidean assumption is not satisfied in distance metric learning, and in dimensionality reduction it is possible to design new algorithms with the distance metric learning goal in mind.

Furthermore, we proved a representer theorem for our framework and the loss function used in the regularization problem is new. It penalizes pairwise differences instead of trying to fit a certain $y$. This is different from previous regularization methods and might be suitable in problems such as multiclass classification.

The squared loss function used in our framework might not be optimal for distance metric learning. However, with the distance metric learning problem transferred to function estimation in an RKHS, one can easily switch to other loss functions and design new algorithms. Better loss functions and a way to automatically determine the parameter settings will be experimented in our future works.

## Acknowledgments

The authors want to thank Wei Liu, Fen Xia and Yanjun Han for helpful discussions. The implementation of tangent distance is kindly provided by Daniel Keysers. This work is supported by the National Grand Fundamental Research 973 Program of China (No.2004CB318103) and the National Science Foundation of China (No. 60673015).

## Appendix

### Proof of Theorem 4

Firstly, we state a lemma which was a partial result of (Sindhwani et al., 2005) and prove two other lemmas.

**Lemma 5** *(Sindhwani et al., 2005) The representer $\tilde{k}(x_i, \cdot)$ of $\tilde{\mathcal{H}}$ can be written as $\tilde{k}(x_i, \cdot) = \sum_{j=1}^{n+m} \beta_j k(x_j, \cdot)$.*

**Lemma 6** *The solution of the optimization problem (4) has a representation $f(x) = \sum_{i=1}^{n+m} \alpha_i k(x_i, x)$. Furthermore, $\alpha$ can be determined by solving the following eigenvalue problem:*

$$\begin{aligned} \min_{\alpha} \quad & \alpha^T (\textstyle\sum_{i,j=1}^{n} w_{ij} (K_i - K_j)(K_i - K_j)^T \\ & \qquad + \lambda (K + KMK))\alpha \\ s.t. \quad & \qquad \alpha^T K^2 \alpha = 1 \end{aligned} \tag{7}$$

*where $K_i$ is the $i$-th column of the Gram matrix $K$.*

**Proof** By standard representer theorems (Kimeldorf & Wahba, 1971; Schölkopf et al., 2001), it is easy to see that the minimizer of the optimization problem (4) has a representation $f(x) = \sum_{i=1}^{n} \theta_i \tilde{k}(x_i, x)$. By Lemma 5, we can rewrite $f(x)$ as $f(x) = \sum_{i=1}^{n+m} \alpha_i k(x_i, x)$. Hence $\|f\|^2_{\tilde{\mathcal{H}}} = \|f\|^2_{\mathcal{H}} + \|Sf\|^2_{\mathcal{V}} = \alpha^T K \alpha + \alpha^T KMK \alpha$. We can get (7) by substituting the expressions of $f(x)$ and $\|f\|^2_{\tilde{\mathcal{H}}}$ into (4). □

**Lemma 7** *For a symmetric $n \times n$ matrix $A$ with rank $n-1$ and $Ae = 0$, we have $AA^+ = A^+A = I - \frac{1}{n}E$.*

**Proof** Suppose that $A$ has an eigenvalue decomposition $A = UDU^T$, then its Moore-Penrose pseudo inverse can be given by $A^+ = UD^+U^T$. Since $A$ has rank $n-1$ and $Ae = 0$, we can partition the decomposition matrices as:

$$A = \begin{bmatrix} U_1 & \frac{1}{\sqrt{n}} e \end{bmatrix} \begin{bmatrix} D_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_1^T \\ \frac{1}{\sqrt{n}} e^T \end{bmatrix},$$

where $D_1 = diag(d_1, \ldots, d_{n-1})$ is the diagonal matrix of the non-zero eigenvalues of $A$. Therefore, $AA^+ = A^+A = U_1 U_1^T$. The lemma then follows from the property of orthogonal matrices: $UU^T = U_1 U_1^T + \frac{1}{n} ee^T = I$. □

**Proof of Theorem 4** From Lemma 6 we have $S(f) = K\alpha$, so we can get $\alpha = K^+ S(f)$. Substituting into (7), we have

$$\begin{aligned} \min_{f} \quad & S(f)^T K^+ (\textstyle\sum_{i,j=1}^{n} w_{ij} (K_i - K_j)(K_i - K_j)^T \\ & \qquad + \lambda (K + KMK)) K^+ S(f). \end{aligned}$$

By Lemma 7, we can get

$$\min_{f} S(f)^T (\sum_{i,j=1}^{n} w_{ij} (s_i - s_j)(s_i - s_j)^T + \lambda (K^+ + N)) S(f),$$

where $N = (I - \frac{1}{n}E) M (I - \frac{1}{n}E)$ and $s_i$ is a vector with 1 in the $i$-th position and 0 in other positions.

Note that $S(f)^T E S(f) = \alpha^T K^T ee^T K \alpha = 0$. And when $X$ is one-dimensional, $\alpha^T K^2 \alpha = S(f)^T S(f) = 1$ is equivalent to $\text{Tr}(X^T X) = 1$. Hence the optimal solutions of (4) and (3) have the correlation $x^* = S(f^*)$. □

## References

Bach, F. R., & Jordan, M. I. (2006). Learning spectral clustering, with application to speech separation. *Journal of Machine Learning Research*, *7*, 1963–2001.

Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, *15*, 1373–1396.

Bengio, Y., Delalleau, O., Roux, N. L., Paiement, J.-F., Vincent, P., & Ouimet, M. (2004). Learning eigenfunctions links spectral embedding and kernel PCA. *Neural Computation*, *16*, 2197–2219.

Bousquet, O., & Hermann, D. J. L. (2003). On the complexity of learning the kernel matrix. *Advances in Neural Information Processing Systems 15.*

Cristianini, N., Kandola, J., Elisseeff, A., & Shawe-Taylor, J. (2002). On kernel-target alignment. *Advances in Neural Information Processing Systems 14.*

Gower, J. C., & Legendre, P. (1986). Metric and euclidean properties of dissimilarities coefficients. *Journal of Classification*, *3*, 5–48.

Ham, J., Lee, D. D., Mika, S., & Schölkopf, B. (2004). A kernel view of the dimensionality reduction of manifolds. *Proceedings of the 21th international conference on machine learning.*

Hoi, S. C. H., Liu, W., Lyu, M. R., & Ma, W.-Y. (2006). Learning distance metrics with contextual constraints for image retrieval. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition.*

Kimeldorf, G. S., & Wahba, G. (1971). Some result on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, *33*, 82–95.

Kwok, J. T., & Tsang, I. W. (2003). Learning with idealized kernels. *Proceedings of the 20th international conference on machine learning.*

Lanckriet, G. R. G., Cristianini, N., Bartlett, P., Ghaoui, L. E., & Jordan, M. I. (2004). Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, *5*, 27–72.

Lebanon, G. (2006). Metric learning for text documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*, 497–508.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*, 2278–2324.

Micchelli, C. A., & Pontil, M. (2005). On learning vector-valued functions. *Neural Computation*, *17*, 177–204.

Scholköpf, B., Herbrich, R., & Smola, A. J. (2001). A generalized representer theorem. *Proceedings of the Annual Conference on Computational Learning Theory.*

Simard, P. Y., LeCun, Y. A., Denker, J. S., & Victorri, B. (1998). Transformation invariance in pattern recognition - tangent distance and tangent propagation. In G. B. Orr and K.-R. Muller (Eds.), *Neural networks: Tricks of the trade.*

Sindhwani, V., Niyogi, P., & Belkin, M. (2005). Beyond the point cloud: from transductive to semi-supervised learning. *Proceedings of the 22th international conference on machine learning.*

Tenenbaum, J., Silva, V. D., & Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*, 2319–2323.

Weinberger, K. Q., Blitzer, J., & Saul, L. K. (2006). Distance metric learning for large margin nearest neighbor classification. *Advances in Neural Information Processing Systems 18.*

Wu, G., Panda, N., & Chang, E. Y. (2005). Formulating context dependent similarity functions. *ACM International Conference on Multimedia (MM).*

Xing, E. P., Ng, A. Y., Jordan, M. I., & Russell, S. (2003). Distance metric learning with applications to clustering with side-information. *Advances in Neural Information Processing Systems 15.*

Zhang, Z. (2003). Learning metrics via discriminant kernels and multidimensional scaling: Toward expected euclidean representation. *Proceedings of the 20th international conference on machine learning.*

Zhang, Z., Kwok, J. T., & Yeung, D.-Y. (2006). Model-based transductive learning of the kernel matrix. *Machine Learning*, *63*, 69–101.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., & Schölkopf, B. (2004). Learning with local and global consistency. *Advances in Neural Information Processing Systems.*