Learning Metrics via Discriminant Kernels and Multidimensional Scaling: Toward Expected Euclidean Representation

Zhihua Zhang ZHZHANG@CS.UST.HK
Department of Computer Science, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

Abstract

Distance-based methods in machine learning and pattern recognition have to rely on a metric distance between points in the input space. Instead of specifying a metric *a priori*, we seek to learn the metric from data via kernel methods and multidimensional scaling (MDS) techniques. Under the classification setting, we define discriminant kernels on the joint space of input and output spaces and present a specific family of discriminant kernels. This family of discriminant kernels is attractive because the induced metrics are Euclidean and Fisher separable, and MDS techniques can be used to find the low-dimensional Euclidean representations (also called feature vectors) of the induced metrics. Since the feature vectors incorporate information from both input points and their corresponding labels and they enjoy Fisher separability, they are appropriate to be used in distance-based classifiers.

1. Introduction

The notion of similarity or dissimilarity plays a fundamental role in machine learning and pattern recognition. For example, distance-based methods such as the k-means clustering algorithm and the nearest neighbor or k-nearest neighbor classification algorithms have to rely on some (dis)similarity measure for quantifying the pairwise (dis)similarity between data points. The performance of a classification (or clustering) algorithm typically depends significantly on the (dis)similarity measure used. Commonly used (dis)similarity measures are based on distance metrics. For many applications, Euclidean distance in the input space is not a good choice. Hence, more complicated distance metrics such as Mahalanobis distance, geodesic distance, chi-square distance and Hausdorff distance, have to be used.

Instead of prespecifying a metric, a promising direction to pursue is metric learning, i.e., to learn an idealized metric from data. More specifically, one wants to embed input points into an idealized (metric) Euclidean space, on which the Euclidean distance accurately reflects the dissimilarity between the corresponding input points. The embedded Euclidean space and points in it are also referred to as a feature space and feature vectors. So the metric learning process can be regarded as an approach to *feature extraction*.

Typically, metric learning consists of two processes. The first is to define an expected distance metric in the feature space with some information from the input space. The second is to embed input points into the feature space with linear or nonlinear mappings. Metric learning methods can be categorized along two dimensions. The first dimension is concerned with what information is used to form dissimilarities. The second dimension is concerned with what techniques are used to calculate the Euclidean embeddings of the dissimilarities.

Along the first dimension, the present literature generally uses one of three basic approaches.

1. *Neighbor Information:* In the unsupervised setting, we take into account the manifold structure of the feature space by preserving local metric information in the input space(Roweis & Saul, 2000; Tenenbaum et al., 2000).

2. *Label Information:* In the supervised setting, the labels of the training points as an important source of information are used to find the Euclidean nature of the feature space (Cox & Ferry, 1993; Webb, 1995; Zhang et al., 2003).

Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

3. *Side Information:* In the semi-supervised setting, given a small set of similar (or dissimilar) pairs in the input space, we learn the metric of the feature space by preserving the geometric properties between the similar (or dissimilar) pairs(Xing et al., 2003). This can be regarded as weak label information.

Along the second dimension, Koontz and Fukunaga (1972) categorized the methods into three approaches.

1. *Iterative Techniques* seek to directly obtain the coordinates, in the feature space, of the input points (Roweis & Saul, 2000; Tenenbaum et al., 2000).
2. *Parametric Techniques* seek to obtain a parameterized regression model from the input space to the feature space by optimizing the model parameters.
3. *Expansion Techniques* are actually a class of parametric techniques, but only the coefficients in the expansion are adjusted(Cox & Ferry, 1993; Webb, 1995; Xing et al., 2003; Zhang et al., 2003).

In general, iterative techniques are used together with neighbor information, while parametric techniques are used together with label information. In this paper, we seek to propose a unifying approach to the metric learning problem according to the above categorizations under the supervised setting. The rest of this paper is organized as follows. In section 2, we propose the basic problem of learning metrics from data. In section 3, we define discriminant kernels and develop a family of discriminant kernels and the induced dissimilarity matrices. In section 4, we discuss Euclidean embedding via MDS techniques. In section 5, our methods are tested on a synthetic dataset and some real datasets. The last section gives some concluding remarks.

2. Problem Formulation

2.1. Euclideanarity and Fisher Separability

First of all, we introduce the following basic definition on Euclideanarity.

Definition 1 (Gower & Legendre, 1986) *An $m \times m$ matrix $\mathbf{A} = [a_{ij}]$ is* Euclidean *if m points P_i ($i = 1, \ldots, m$) can be embedded in an Euclidean space such that the Euclidean distance between P_i and P_j is a_{ij}.*

In this paper, we also refer to $\mathbf{A}$ with elements a_{ij} as being Euclidean if and only if a matrix with $a_{ij}^{\frac{1}{2}}$ is Euclidean. The following theorem provides the conditions for matrix $\mathbf{A}$ to be Euclidean.

Theorem 1 (Gower & Legendre, 1986) *The matrix $\mathbf{A} = [a_{ij}]$ is Euclidean if and only if $\mathbf{H}'\mathbf{B}\mathbf{H}$ is positive semi-definite, where $\mathbf{B}$ is the matrix with elements $-\frac{1}{2}a_{ij}^2$ and $\mathbf{H} = (\mathbf{I} - \mathbf{1}_m\mathbf{s}')$ is a centering matrix where $\mathbf{I}$ is the identity matrix, $\mathbf{1}_m$ is the $m \times 1$ vector $(1, 1, \ldots, 1)'$ and $\mathbf{s}$ is a vector satisfying $\mathbf{s}'\mathbf{1}_m = 1$.*

Since Euclidean distance satisfies the triangle inequality, a necessary condition for $\mathbf{A}$ to be Euclidean is that it be metric; that this is not also a sufficient condition (Gower & Legendre, 1986).

Inspired by the Fisher discriminant criterion, we define the notion of *Fisher Separability* here, as follows.

Definition 2 *An $m \times m$ matrix $\mathbf{A} = [a_{ij}]$ is* Fisher separable *if $a_{ij} < a_{lk}$ where two points P_i and P_j corresponding to a_{ij} belong to the same class, and points P_l and P_k corresponding to a_{lk} belong to different classes.*

In other words, a dissimilarity matrix is Fisher separable if its between-class dissimilarity is always larger than its within-class dissimilarity.

2.2. Kernel Trick for Dissimilarity Matrices

Recently, kernel-based methods (Schölkopf & Smola, 2002; Vapnik, 1995) are increasingly being applied to data and information processing due to their conceptual simplicity and theoretical potentiality. Kernel methods work over the feature space $\mathcal{F}$, which is related to an input space $\mathcal{I}$ by a nonlinear map φ. In order to compute inner-products of the form $\varphi(\mathbf{a})'\varphi(\mathbf{b})$, we use a kernel function $k(\mathbf{a}, \mathbf{b})$ to represent

$$k : \mathcal{I} \times \mathcal{I} \to \mathbb{R}, \qquad k(\mathbf{a}, \mathbf{b}) = \varphi(\mathbf{a})'\varphi(\mathbf{b}),$$

which allows us to compute the value of the inner-product in $\mathcal{F}$ without carrying out the map φ. This is sometimes referred to as the *kernel trick*. In most cases, we pay much attention to positive definite kernels. If $\mathcal{I}$ is a finite set ($\mathcal{I} = \{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$, say), then k is positive definite if and only if the $m \times m$ *Gram matrix* (also called the *kernel matrix*) $\mathbf{K} = [k(\mathbf{a}_i, \mathbf{a}_j)]$ is positive semi-definite.

In the feature space, the squared distance σ_{ij}^2 between feature vectors $\varphi(\mathbf{a}_i)$ and $\varphi(\mathbf{a}_j)$ can be defined as

$$\sigma_{ij}^2 = \|\varphi(\mathbf{a}_i) - \varphi(\mathbf{a}_j)\|^2 = k_{ii} + k_{jj} - 2k_{ij}, \quad (1)$$

where $k_{ij} = k(\mathbf{a}_i, \mathbf{a}_j)$. Let $\mathbf{\Delta} = [\sigma_{ij}^2]$ where σ_{ij}^2 is defined in (1). $\mathbf{\Delta}$ can be expressed in matrix form as,

$$\mathbf{\Delta} = \mathbf{k}\mathbf{1}_m' + \mathbf{1}_m\mathbf{k}' - 2\mathbf{K}, \quad (2)$$

where $\mathbf{k} = (k_{11}, \ldots, k_{mm})'$. Using Theorem 1 and

$$-\frac{1}{2}\mathbf{H}'\mathbf{\Delta}\mathbf{H} = \mathbf{H}'\mathbf{K}\mathbf{H},$$

we have

Corollary 1 *The dissimilarity matrix $\mathbf{\Delta}$ is Euclidean if and only if k is a positive definite kernel.*

2.3. Basic Problem

Suppose we have an input set $\mathcal{X} \subset \mathbf{R}^d$ and an output set $\mathcal{T} = \{1, 2, \ldots, c\}$ of all c possible class (target) labels. We are given a training set $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{T}$ with n points $\{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_n, t_n)\}$, where $t_i = r$ if point i belongs to class r.[1] Here and later, by $\hat{\mathbf{x}} \in \mathbf{R}^l$ we denote the feature vector corresponding to the input point $\mathbf{x}$, and by d_{ij} and $\hat{d}_{ij}$ we denote the distances between points i and j in the input space and in the feature space, respectively. For simplicity, from now on, we always refer to $\hat{d}_{ij}$'s and $\hat{\mathbf{D}} = [\hat{d}_{ij}]$ as dissimilarities and the *dissimilarity matrix*.

Simply speaking, metric learning wants to obtain $\hat{\mathbf{x}}$'s such that the Euclidean distance between $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}_j$ approximates $\hat{d}_{ij}$, and consists of two processes: defining $\hat{\mathbf{D}}$ and calculating $\hat{\mathbf{x}}$'s. Since the performance of metric learning methods depends severely on the $\hat{\mathbf{D}}$, we focus our main attention on the first process of metric learning in this paper.

Among the three techniques of obtaining $\hat{\mathbf{x}}$'s, MDS techniques (Borg & Groenen, 1997) are widely used. For example, we generally employ classical MDS to develop iterative techniques (Roweis & Saul, 2000; Tenenbaum et al., 2000) and use metric or nonmetric MDS models to develop parametric techniques (Cox & Ferry, 1993; Webb, 1995; Zhang et al., 2003). If dissimilarity matrices are Euclidean or metric, we will be able to obtain efficient implementations of the MDS methods.

As analyzed in Section 2.2, the dissimilarity matrices induced from kernel matrices are Euclidean. On the other hand, Fisher separability is a very useful criterion for discriminant analysis and clustering. In most of the existent kernel methods, although the feature vectors in the feature space are more likely to be linearly separable than the input points in the input space, the induced dissimilarity matrices do not necessarily satisfy our expected Fisher separability.

In this paper, our concerned problem is on *how to construct dissimilarity matrices with both Euclideanarity and Fisher separability.* We will address this problem with the kernel trick.

[1] In this paper we only consider the case in which each point is assumed to belong to only one class.

2.4. Related Work

Some work in the literature is related to our work. In the existent literature, the goal of using neighbor information, label information or side information is to follow Fisher separability to define $\hat{\mathbf{D}} = [\hat{d}_{ij}]$. For example, Cox and Ferry (1993; 1995; 2003) seek to increase the between-class dissimilarity and decrease the within-class dissimilarity. However, the dissimilarity matrices do not still satisfy the Fisher separability. Moreover, the dissimilarity matrices are not guaranteed to be metric and Euclidean. Zhang et al. (2003) defined the dissimilarity matrix with the Fisher separability. However, ones have not proven it to be Euclidean.

Recently, Cristianini et al. (2002) considered the relationship between the input kernel matrix and the target kernel matrix, and presented the notion of the *alignment* of two kernel matrices. Based on this notion, some methods of learning the kernel matrix have been successively presented in the transductive or inductive setting (Cristianini et al., 2002; Lanckriet et al., 2002; Bousquet & Herrmann, 2003; Kandola et al., 2002a; Kandola et al., 2002b). Our work differs from that of Cristianini et al. (2002) in that we develop a new kernel matrix via the input kernel matrix and the output kernel matrix, while they seek to measure the correlation between these two kernel matrices. On the other hand, our work can also be regarded as a parametric model of learning the kernel matrix because we directly obtain the coordinates of the feature vectors, and the inner products of these coordinates form an idealized kernel matrix. Compared with the above methods, the computational complexity of our model is lower.

3. Discriminant Kernels

3.1. Tensor Product and Direct Sum Kernels

There exist a number of different methods for constructing new kernels from existing ones (Haussler, 1999). In this section, we consider two such examples. Given $x_1, x_2 \in \mathcal{X}$ and $u_1, u_2 \in \mathcal{U}$. If k_1, k_2 are kernels defined on $\mathcal{X} \times \mathcal{X}$ and $\mathcal{U} \times \mathcal{U}$ respectively, then their tensor product (Wahba, 1990), $k_1 \otimes k_2$, defined on $(\mathcal{X} \times \mathcal{U}) \times (\mathcal{X} \times \mathcal{U})$ as

$$k_1 \otimes k_2((x_1, u_1), (x_2, u_2)) = k_1(x_1, x_2)k_2(u_1, u_2),$$

is called the *tensor product kernel*.

Similarly, their direct sum, $k_1 \oplus k_2$, defined on $(\mathcal{X} \times$

$\mathcal{U}) \times (\mathcal{X} \times \mathcal{U})$ as

$$k_1 \oplus k_2((x_1, u_1), (x_2, u_2)) = k_1(x_1, x_2) + k_2(u_1, u_2),$$

is called the *direct sum kernel*.

3.2. Definition of Discriminant Kernels

We observe the training set $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{T}$. In kernel methods, kernels are mostly defined on $\mathcal{X} \times \mathcal{X}$. In this paper, our idea is to define kernels on the training set $\mathcal{D}$ using label information or side information. We define the respective tensor product and direct sum kernels on $(\mathcal{X} \times \mathcal{T}) \times (\mathcal{X} \times \mathcal{T})$ as,

$$(k_1 \otimes k_2)((\mathbf{x}_i, t_i), (\mathbf{x}_j, t_j)) = k_1(\mathbf{x}_i, \mathbf{x}_j) k_2(t_i, t_j), \quad (3)$$

and

$$(k_1 \oplus k_2)((\mathbf{x}_i, t_i), (\mathbf{x}_j, t_j)) = \alpha_1 k_1(\mathbf{x}_i, \mathbf{x}_j) + \alpha_2 k_2(t_i, t_j), \quad (4)$$

where $\alpha_1, \alpha_2 \geq 0$ are mixing weights. We call the kernels defined above *discriminant kernels*. We know that in the joint space containing the training set, the input points and their corresponding labels have different Euclidean characteristics. For example, the input points are generally continuous variables while the labels are discrete variables. Obviously, the Euclidean distance on this joint space is unexpected. Using the discriminant kernels, our goal here is to embed the joint space into a feature space, on which the Euclidean distance is idealized.

For convenience, by $(k_1 \otimes k_2)_{ij}$ and $(k_1 \oplus k_2)_{ij}$ we denote $(k_1 \otimes k_2)((\mathbf{x}_i, t_i), (\mathbf{x}_j, t_j))$ and $(k_1 \oplus k_2)((\mathbf{x}_i, t_i), (\mathbf{x}_j, t_j))$, respectively. Thus, we can compute the squared distances between feature vectors embedded with the discriminant kernels

$$\hat{d}_{ij}^2 = (k_1 \otimes k_2)_{ii} + (k_1 \otimes k_2)_{jj} - 2(k_1 \otimes k_2)_{ij}, \quad (5)$$

or

$$\hat{d}_{ij}^2 = (k_1 \oplus k_2)_{ii} + (k_1 \oplus k_2)_{jj} - 2(k_1 \oplus k_2)_{ij}. \quad (6)$$

Corollary 2 *If k_1 and k_2 are positive definite kernels, then the matrix $\hat{\mathbf{D}} = [\hat{d}_{ij}]$ where $\hat{d}_{ij}$'s are defined by (5) or (6), is Euclidean.*

Proof. We know that both the discriminant kernels defined by (3) and (4) are positive definite since k_1, k_2 are positive definite. By Corollary 1, $\hat{\mathbf{D}}$ is Euclidean. □

3.3. Construction of Discriminant Kernels

Once the kernels k_1 and k_2 have been given, we can obtain the discriminant kernels using the tensor product or the direct sum. Here, we choose the Gaussian kernels or correlation kernels[2] as k_1,

$$k_1(\mathbf{x}_i, \mathbf{x}_j) = g_{ij}(\beta) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\beta}\right), \quad (7)$$

$$k_1(\mathbf{x}_i, \mathbf{x}_j) = \rho_{ij}^q = \frac{(\mathbf{x}_i' \mathbf{x}_j)^q}{(\|\mathbf{x}_i\| . \|\mathbf{x}_j\|)^q}, \quad (8)$$

and the trivial kernel [3] (Cristianini et al., 2002)

$$k_2(t_i, t_j) = \delta(t_i, t_j),$$

as k_2. In the above equations $\beta > 0$ is a scaling constant, q is the degree of the polynomial, and $\delta(t_i, t_j)$, the Kronecker delta function, is defined such that $\delta(t_i, t_j) = 1$ when $t_i = t_j$ and 0 otherwise. We set $\alpha_1 = \alpha_2 = \frac{1}{2}$, the discriminant kernels then become

$$(k_1 \otimes k_2)_{ij} = \begin{cases} g_{ij}(\beta) & t_i = t_j \\ 0 & t_i \neq t_j \end{cases}, \quad (9)$$

$$(k_1 \oplus k_2)_{ij} = \begin{cases} \frac{1}{2} g_{ij}(\beta) + \frac{1}{2} & t_i = t_j \\ \frac{1}{2} g_{ij}(\beta) & t_i \neq t_j \end{cases}, \quad (10)$$

for the Gaussian kernels, and

$$(k_1 \otimes k_2)_{ij} = \begin{cases} \rho_{ij}^q & t_i = t_j \\ 0 & t_i \neq t_j \end{cases}, \quad (11)$$

$$(k_1 \oplus k_2)_{ij} = \begin{cases} \frac{1}{2} \rho_{ij}^q + \frac{1}{2} & t_i = t_j \\ \frac{1}{2} \rho_{ij}^q & t_i \neq t_j \end{cases}, \quad (12)$$

for the correlation kernels.

From the definition of the trivial kernel on the label set $\mathcal{T}$, the trivial kernel works well if we are given only the pairwise side information of the labels instead of the values of the labels.

3.4. Dissimilarity Matrices

We regard the squared distances between feature vectors as our concerned dissimilarities. Corresponding to the discriminant kernels in (9)-(12), we obtain the dissimilarities and denote them as

$$\delta_{ij}^{(1)} = \begin{cases} 2 - 2g_{ij}(\beta) & t_i = t_j \\ 2 & t_i \neq t_j \end{cases}, \quad (13)$$

$$\delta_{ij}^{(2)} = \begin{cases} 1 - g_{ij}(\beta) & t_i = t_j \\ 2 - g_{ij}(\beta) & t_i \neq t_j \end{cases}, \quad (14)$$

$$\delta_{ij}^{(3)} = \begin{cases} 2 - 2\rho_{ij}^q & t_i = t_j \\ 2 & t_i \neq t_j \end{cases}, \quad (15)$$

$$\delta_{ij}^{(4)} = \begin{cases} 1 - \rho_{ij}^q & t_i = t_j \\ 2 - \rho_{ij}^q & t_i \neq t_j \end{cases}. \quad (16)$$

[2] It can be regarded as first normalizing $\mathbf{x}_i$ by $\mathbf{y}_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|}$ and then using the polynomial kernels $(\mathbf{y}_i' \mathbf{y}_j)^q$.

[3] The feature map is $\varphi(t) = (\delta(t, 1), \ldots, \delta(t, c))'$. This finding has been presented by Bach and Jordan (2003). This is equivalent to directly setting $\mathbf{t} = (\delta(t, 1), \ldots, \delta(t, c))'$ and $k_2(\mathbf{t}_i, \mathbf{t}_j) = \mathbf{t}_i' \mathbf{t}_j$.

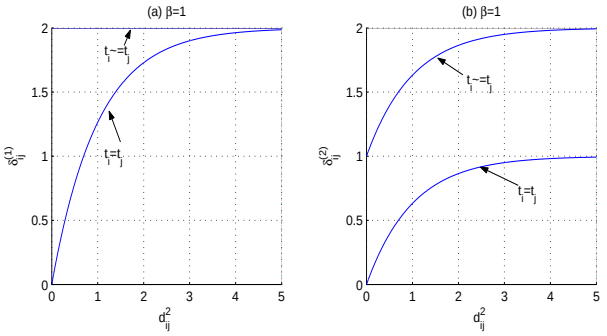


Figure 1. Dissimilarities defined by (13) and (14).

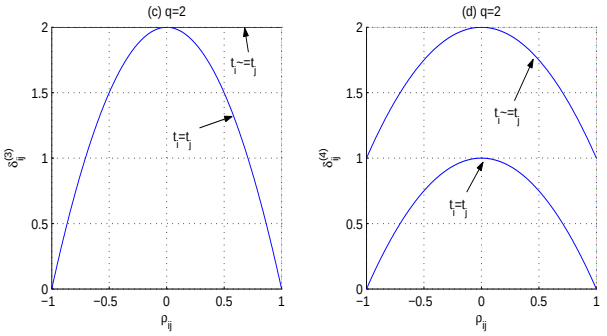


Figure 2. Dissimilarities defined by (15) and (16).

We can see that $\delta_{ij}^{(1)}$ and $\delta_{ij}^{(2)}$ are related to distances d_{ij}'s in the input space, while $\delta_{ij}^{(3)}$ and $\delta_{ij}^{(4)}$ are related to correlation coefficients ρ_{ij}'s in the input space. Figures 1 and 2 illustrate the propositions of these dissimilarities, where $\beta = 1$ or $q = 2$. As $-1 \leq \rho_{ij} \leq 1$, it is easy to obtain the following theorem

Theorem 2 *The dissimilarity matrices* $\hat{\mathbf{D}}^{(k)} = [\delta_{ij}^{(k)}]$ *for* $k = 1, \ldots, 4$*, where* $\beta > 0$ *or* q *is a positive even number, are Euclidean and Fisher separable.*

Theorem 2 shows that the problem given in Section 2.3 has been successfully resolved by using our discriminant kernels.

4. Euclidean Embedding with Metric MDS

Notice that the resultant dissimilarities $\hat{d}_{ij}$'s incorporate information from both the input points ($\mathbf{x}_i$'s) and their corresponding class labels (t_i's). Moreover, they possess the property of Fisher separability. They are appropriate to be used in distance-based classifiers. The subsequent task is then to find the embeddings, i.e., feature vectors $\{\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_n\} \in \mathbb{R}^l$ such that the inter-point distances are equal to (or approximative) $\hat{d}_{ij}$'s defined in the above section. Because $\hat{\mathbf{D}}$ is Euclidean, a natural solution can be easily obtained by using the classical MDS, i.e., an iterative technique. However in the classification scenario, this may be intractable because for new points with unknown labels, the problem then is on how to determine $\hat{d}_{ij}$ in the first process. Fortunately, using parametric techniques (Koontz & Fukunaga, 1972; Cox & Ferry, 1993; Webb, 1995), we can avoid this problem. Here we employ an expansion technique proposed by Webb (1995).

4.1. Expansion Techniques

Denote the mapping from the original input space $\mathbb{R}^d$ to the embedded Euclidean space $\mathbb{R}^l$ by $\mathbf{f} = (f_1, \ldots, f_l)'$. We assume that each f_i is a linear combination of p basis functions:

$$f_i(\mathbf{x}; \mathbf{W}) = \sum_{j=1}^{p} w_{ji} \phi_j(\mathbf{x}), \tag{17}$$

where $\mathbf{W} = [w_{ji}]_{p \times l}$ contains the free parameters, and the basis functions $\phi_j(\mathbf{x})$'s can be linear or nonlinear. The regression mapping (17) can be written in matrix form as

$$\mathbf{f}(\mathbf{x}; \mathbf{W}) = \mathbf{W}' \boldsymbol{\phi}(\mathbf{x}),$$

where $\boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_p(\mathbf{x}))'$. Letting $\mathbf{Y} = [\mathbf{f}(\mathbf{x}_1) \ldots, \mathbf{f}(\mathbf{x}_n)]'$ the resultant configuration obtained by the regression mapping, we seek to minimize the squared error as

$$e^2(\mathbf{W}) = \sum_{i=1}^{n} \sum_{j=1}^{n} (\hat{d}_{ij} - q_{ij}(\mathbf{W}))^2, \tag{18}$$

where $q_{ij}(\mathbf{W}) = \|\mathbf{W}'(\boldsymbol{\phi}_i - \boldsymbol{\phi}_j)\|$ with $\boldsymbol{\phi}_i = \boldsymbol{\phi}(\mathbf{x}_i)$.

The so-called expansion techniques seek to minimize e^2 w.r.t. $\mathbf{W}$, given the basis functions $\boldsymbol{\phi}(\mathbf{x})$.

4.2. Iterative Majorization Algorithm

Here the iterative majorization algorithm (Borg & Groenen, 1997) is used to address the above expansion model. The procedure for obtaining $\mathbf{W}$ can be summarized as follows:

1. Set $t = 0$ and initialize $\mathbf{W}^{(t)}$.
2. Set $\mathbf{V} = \mathbf{W}^{(t)}$.
3. Update $\mathbf{W}^{(t)}$ to $\mathbf{W}^{(t+1)}$, where $\mathbf{W}^{(t+1)}$ is the solution of $\mathbf{W}$ in equation as

$$\mathbf{W} = \mathbf{B}^{+}\mathbf{C}(\mathbf{V})\mathbf{V}, \tag{19}$$

where $\mathbf{B} = \sum_{i,j}(\boldsymbol{\phi}_i - \boldsymbol{\phi}_j)(\boldsymbol{\phi}_i - \boldsymbol{\phi}_j)'$, $\mathbf{B}^+$ is the Moore-Penrose inverse[4] of $\mathbf{B}$, and $\mathbf{C}(\mathbf{V}) = \sum_{i,j} c_{ij}(\mathbf{V})(\boldsymbol{\phi}_i - \boldsymbol{\phi}_j)(\boldsymbol{\phi}_i - \boldsymbol{\phi}_j)'$ with

$$c_{ij}(\mathbf{V}) = \begin{cases} \frac{d_{ij}(\mathbf{X})}{q_{ij}(\mathbf{V})} & q_{ij}(\mathbf{V}) > 0, \\ 0 & q_{ij}(\mathbf{V}) = 0. \end{cases} .$$

4. Check for convergence. If not converged, set $t = t + 1$ and go to step 2; otherwise stop.

The details of the method can be found in (Zhang et al., 2003). The functions $\boldsymbol{\phi}(\mathbf{x})$ can be chosen to be different linear or nonlinear functions.

5. Experiments

In the first two experiments, we aim at dimensionality reduction and data visualization, and concern finding suitable low-dimensional features with which the inter-point distances approximate as well as possible the dissimilarities. We shall use the Gaussian radial basis functions as

$$\phi_j(\mathbf{x}) = \exp\left\{-\frac{\| \mathbf{x} - \mathbf{m}_j \|^2}{\beta}\right\},$$

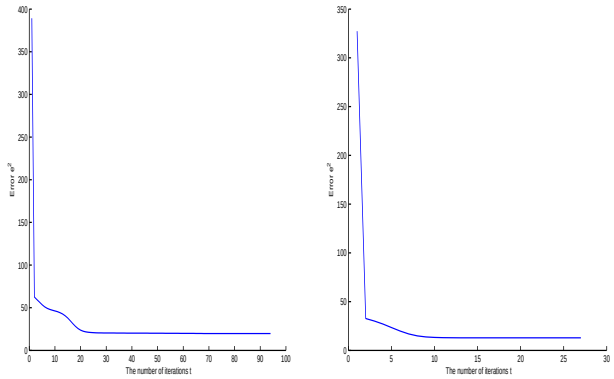
and set $l = 2$. In the third experiment, we apply our model to classification problems on the real data, and use $l = p = q$ and $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$. The width β is set to be the average distance of the labelled points to their class means.

5.1. Synthetic Data

This data set consists of 300 two-dimensional points partitioned into two ring-like classes, each with 150 points (Figure 3(a)). The training subset consists of 20 points selected randomly from each class, while the remaining points form the test set. We take 10 basis functions and randomly sample 5 points from each class of the training subset as centers of the basis functions.

The experiments were run for 10 random initializations of $\mathbf{W}^{(0)}$ for the different dissimilarities $\delta_{ij}^{(1)}$, $\delta_{ij}^{(2)}$, $\delta_{ij}^{(3)}$ and $\delta_{ij}^{(4)}$, respectively. We find that for $\delta_{ij}^{(1)}$, $\delta_{ij}^{(2)}$, $\delta_{ij}^{(3)}$ and $\delta_{ij}^{(4)}$, the respective experimental results are almost the same with respect to different initializations. We show the results for one initialization in Figure 3, where the the feature vectors corresponding to all input points have been obtained via the above expansion technique after the iterative majorization algorithm has converged. We can see that when using $\delta_{ij}^{(2)}$, $\delta_{ij}^{(3)}$ and $\delta_{ij}^{(4)}$, especially using $\delta_{ij}^{(2)}$ and $\delta_{ij}^{(4)}$, the obtain feature vectors are expected. So we feel that using the direct sum is more effective than using the tensor product for constructing the discriminant kernels.

[4] $\mathbf{B}^+$ is used instead of $\mathbf{B}^{-1}$ because $\mathbf{B}$ may not be of full rank.



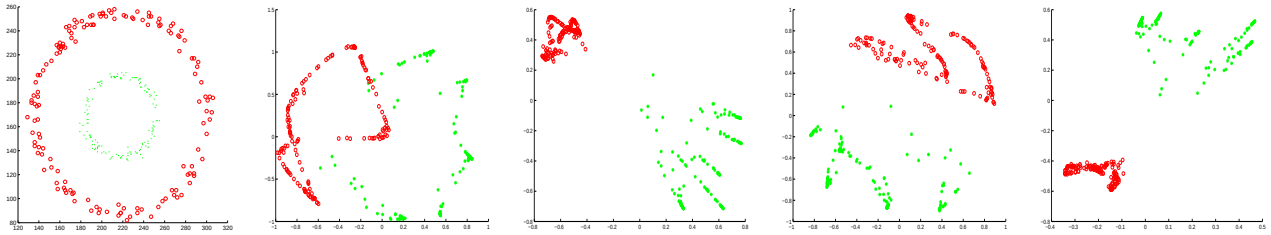
(a) for $\delta_{ij}^{(2)}$ (b) for $\delta_{ij}^{(4)}$

Figure 4. The training errors in running the iterative majorization procedure for partially labelled input points.

5.2. Iris Data

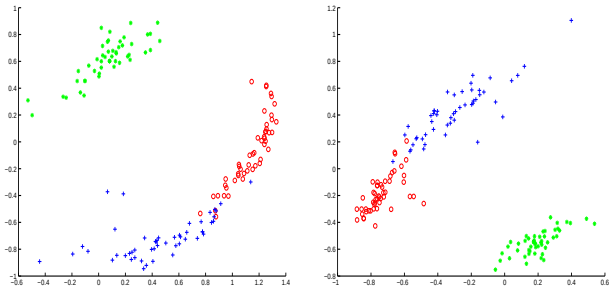
In this section, we test our methods on the *Iris* data set with 3 classes, each of which consists of 50 four-dimensional points, and use the dissimilarities $\delta_{ij}^{(2)}$ and $\delta_{ij}^{(4)}$. Firstly, we use only a subset of points to learn the metric and then perform the Euclidean embeddings of the remaining points using the above expansion technique. The subset consists of 10 points randomly sampled from each class. Figure 4 and Figure 5 plot the decrease in e^2 versus the number of iterations in running the iterative majorization procedure and the the obtained feature vectors, respectively. We chose 6 basis functions and randomly sample 2 points from each class of the training subset as the centers of the basis functions. As we can be see, the convergence of the iterative majorization is very fast, requiring less than 20 iterations. We also find that the feature vectors in two-dimensional space are more separable than the input points in the original four-dimensional space. However, our expected goal is still not achieved, i.e., we do not get that the between-class dissimilarity is larger than the within-class one.

Secondly, we use all 150 points to learn the metric, and use classical MDS and the above expansion technique, respectively, to obtain the feature vectors. Figure 6 shows the two-dimensional feature vectors obtained



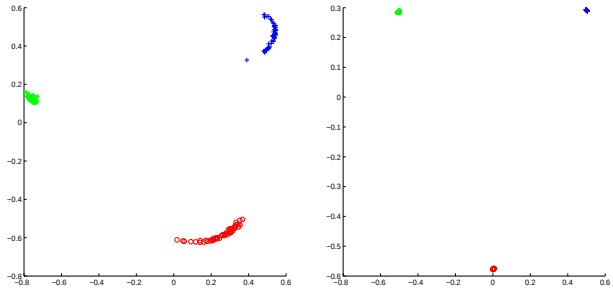
(a) (b) for $\delta_{ij}^{(1)}$ (c) for $\delta_{ij}^{(2)}$ (d) for $\delta_{ij}^{(3)}$ (e) for $\delta_{ij}^{(4)}$

Figure 3. (a) Original input points; and (b)-(e) the feature vectors using discriminant kernels.



(a) for $\delta_{ij}^{(2)}$ (b) for $\delta_{ij}^{(4)}$

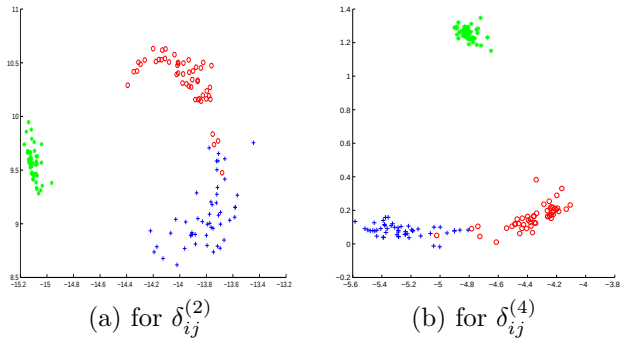
Figure 5. The feature vectors using the expansion technique and partially labelled input points.



(a) for $\delta_{ij}^{(2)}$ (b) for $\delta_{ij}^{(4)}$

Figure 6. The feature vectors using classical MDS for fully labelled input points.

by classical MDS. Figure 7 shows the two-dimensional feature vectors obtained by the expansion technique, where we use 24 basis functions and randomly sample the centers of these basis functions from the input points. Clearly, classical MDS can obtain the idealized feature vectors, whereas the expansion technique can not. This can be attributed to the approximation ability of the used regression model in (17) and the local convergence of the iterative majorization algorithm. These problems are related to the second process of metric learning. Since this paper focuses mainly on the first process, here we do not give more discussions on the second process. This process will be further addressed in the future.



(a) for $\delta_{ij}^{(2)}$ (b) for $\delta_{ij}^{(4)}$

Figure 7. The feature vectors using the expansion technique for fully labelled input points.

5.3. Applications to Real Datasets

In this section, we perform experiments on five benchmark datasets from the UCI repository[5](Pima Indians diabetes, soybean, wine, Wisconsin breast cancer and ionosphere). The distance metric is learned using a small subset of the labelled points, whose sizes are detailed in (Zhang et al., 2003), while the remaining points are then used for testing. We apply the nearest mean (NM) and nearest neighbor (NN) classifiers on the input space and the feature space, respectively. The experiments were run for 10 random initializations of $\mathbf{W}^{(0)}$ with the dissimilarities $\delta_{ij}^{(2)}$, and the average of classification results are reported in Table 1. It can be seen that the results on the feature space almost always outperform the results on the input space.

[5]http://www.ics.uci.edu/mlearn/MLRepository.html

6. Conclusion

In order to obtain idealized Euclidean representations of the input points, we have discussed the metric learn-

Table 1. Classification results on the UCI datasets (# points correctly classified / # points for testing).

data set	input space		feature space	
	NM	NN	NM	NN
pima	463/638	432/638	477/638	428/638
soybean	36/37	35/37	37/37	37/37
wine	86/118	77/118	113/118	115/118
breast	430/469	420/469	448/469	451/469
ionosphere	159/251	212/251	201/251	224/251

ing methods by means of the kernel methods and MDS techniques. In the classification scenario, we defined discriminant kernels on the joint space of input and output spaces, and presented a specific family of the discriminant kernels. This family of the discriminant kernels is attractive because the induced metrics are Euclidean and Fisher separable.

Acknowledgments

The author would like to thank Dit-Yan Yeung and James T. Kwok for fruitful discussions. Many thanks to the anonymous reviewers for their useful comments.

References

Bach, F. R., & Jordan, M. I. (2003). Learning graphical models with Mercer kernels. *Advances in Neural Information Processing Systems 15*. Cambridge, MA: MIT Press.

Borg, I., & Groenen, P. (1997). *Modern multidimensional scaling*. New York: Springer-Verlag.

Bousquet, O., & Herrmann, D. J. L. (2003). On the complexity of learning the kernel matrix. *Advances in Neural Information Processing Systems 15*. Cambridge, MA: MIT Press.

Cox, T. F., & Ferry, G. (1993). Discriminant analysis using non-metric multidimensional scaling. *Pattern Recognition*, *26*, 145–153.

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

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

Haussler, D. (1999). *Convolution kernels on discrete structures* (Technical Report UCSC-CRL-99-10). Department of Computer Science, University of California at Santa Cruz.

Kandola, J., Shawe-Taylor, J., & Cristianini, N. (2002a). *On the extensions of kernel alignment* (Technical Report 2002-120). NeuroCOLT.

Kandola, J., Shawe-Taylor, J., & Cristianini, N. (2002b). *Optimizing kernel alignment over combinations of kernels* (Technical Report 2002-121). NeuroCOLT.

Koontz, W. L. G., & Fukunaga, K. (1972). A nonlinear feature extraction algorithm using distance information. *IEEE Transactions on Computers*, *21*, 56–63.

Lanckriet, G. R. G., Cristianini, N., Ghaoui, L. E., Bartlett, P., & Jordan, M. I. (2002). Learning the kernel matrix with semi-definite programming. *The 19th International Conference on Machine Learning*.

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*, 2323–2326.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*. The MIT Press.

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

Vapnik, V. (1995). *The nature of statistical learning theory*. New York: Springer-Verlag.

Wahba, G. (1990). *Spline models for observational data*. Philadelphia: SIAM.

Webb, A. R. (1995). Multidimensional scaling by iterative majorization using radial basis functions. *Pattern Recognition*, *28*, 753–759.

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

Zhang, Z., Kwok, J. T., & Yeung, D. Y. (2003). *Parametric distance metric with label information* (Technical Report HKUST-CS03-02). Department of Computer Science, Hong Kong University of Science and Technology.