2009 10th International Conference on Document Analysis and Recognition

Pattern classification on local metric structure

Yoshikazu WASHIZAWA
Brain Science Institute, RIKEN.
washizawa@brain.riken.jp

Abstract

A metric is an important concept in pattern classification problems. Many metrics have been applied to pattern classification problems, e.g., the Mahalanobis distance or shift-invariant distance. However, a metric is not uniform in whole domain, in other words, structure of patterns are different in each local domain. Several approaches that utilize such local structure have been proposed.

In this paper, we systematize them and propose a framework to describe patterns by a d-dimensional vector and local metric matrix at the point. Then, we introduce two distance measurements to this framework. Experimental results demonstrate advantages of the proposed methods.

1. Introduction

In pattern classification problems, input data is often given as a d-dimensional vector in the Euclidean space, and classifiers are constructed as a function on the Euclidean space. However, non-vector type input data may be obtained in many problems. For example, from an image sequence, we can get several images as a single pattern. If the number of images are not fixed, we often extract feature as a vector using averaging. Then the information of the second order statistics is not utilized. Another example is so-called transformation invariance pattern recognition such as a tangent distance [7]. We sometimes know local property of input patterns as the prior-knowledge. In character recognition problems, transformations such as rotation, scaling and shifting are invariant for features. Such transforms draw smooth surface in d-dimensional Euclidean space. The tangent distance method considers the tangent space of the surface at the point of the input vector, and defines the dissimilarity as the minimum distance of the tangent spaces.

Furthermore, recently, many supervised and unsupervised learning methods for metric learning have been proposed [2, 10, 13, 8]. They estimate local metric matrix from samples, and each pattern is represented by a set of a d-dimensional vector and a metric matrix. However, they use one metric matrix to estimate distance between two patterns.

In this paper, we focus on the case that input patterns are given as a set of a d-dimensional vector $\boldsymbol{x}$ and a $d \times d$ matrix $\boldsymbol{Y}$, $(\boldsymbol{x}, \boldsymbol{Y}) \in \mathbb{R}^d \times \mathbb{R}^{d \times d}$, where $\times$ denotes the direct product. The matrix $\boldsymbol{Y}$ is a metric tensor at the point $\boldsymbol{x}$. $\boldsymbol{Y}$ can be characterized by the second order statistics of the sequence or prior-knowledge of patterns. In the case of images sequence, $\boldsymbol{x}$ is a mean vector, and $\boldsymbol{Y}$ is an inverse of variance-covariance matrix. $\boldsymbol{Y}$ can also be obtained from a projection matrix onto the tangent space.

In order to make a classifier on the direct product space, $\mathbb{R}^d \times \mathbb{R}^{d \times d}$, we shall define the distance (or similarity) between two patterns $(\boldsymbol{x}_1, \boldsymbol{Y}_1)$, $(\boldsymbol{x}_2, \boldsymbol{Y}_2)$. Section 2 introduces two conventional methods to define the distance. One is the mutual subspace method and another is the tangent distance method, and we show they are not appropriate measurements for universal cases. In Section 4, we propose new two measurements named metric interpolation function (MIF) and highest valley function (HVF). The optimization problem of HVF is reduced to a semi-definite programing problem (SDP). It is known that SDP is solved in polynomial order, however, it is still high computational cost. We therefore, propose a fast optimization method based on linear search problem. Section 5 presents experimental results to demonstrate the advantages of the proposed method. Section 6 concludes the paper.

2. Classical methods

2.1. Tangent distance

Tangent distance method includes two steps; 1) obtain tangent space of an input vector $\boldsymbol{x}$; 2) calculate the similarity of two input patterns.

For the step 1, several transformed patterns that is called tangent vectors is obtained from an input vectors, we denote the tangent vectors by $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$ $(m < d)$. Then the m-dimensional subspace spanned by $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$ is defined as the tangent space of the input vector. In [7], seven kinds of transforms, X-translation, Y-translation, rotation,

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.151

scaling, parallel hyperbolic transformation, diagonal hyperbolic transformation and thickening are proposed. Suppose that the transformed vector $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$ are linear independent. Let $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m$ be orthonormal basis of the tangent space, and $\boldsymbol{U} = [\boldsymbol{u}_1 \ldots \boldsymbol{u}_m] \in \mathbb{R}^{d \times m}$. The input vector is transformed to a linear variety $\mathcal{X}$,

$$\mathcal{X} = \{\boldsymbol{z} | \boldsymbol{z} = \boldsymbol{x} + \boldsymbol{U}\boldsymbol{\alpha},\ \boldsymbol{\alpha} \in \mathbb{R}^m\}. \tag{1}$$

In the next step, we calculate the distance between two input patterns, $\boldsymbol{x}_1$, $\boldsymbol{x}_2$. Let $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ be corresponding matrices of orthonormal basis of the tangent spaces, and let $\mathcal{X}_1$ and $\mathcal{X}_2$ corresponding to linear verities. The distance between $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ is defined as the minimum distance of the linear varieties;

$$\begin{aligned} D(\mathcal{X}_1, \mathcal{X}_2) &= \min_{\boldsymbol{z}_1 \in \mathcal{X}_1, \boldsymbol{z}_2 \in \mathcal{X}_2} \|\boldsymbol{z}_1 - \boldsymbol{z}_2\|^2 \\ &= \min_{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2} \|(\boldsymbol{x}_1 + \boldsymbol{U}_1\boldsymbol{\alpha}_1) - (\boldsymbol{x}_2 + \boldsymbol{U}_2\boldsymbol{\alpha}_2)\|^2, \end{aligned} \tag{2}$$

where $\|\cdot\|$ denotes the Euclidean norm. The minimum solution is given by $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ such that

$$\begin{aligned} (\boldsymbol{I} - \boldsymbol{U}_1^\top \boldsymbol{U}_2 \boldsymbol{U}_2^\top \boldsymbol{U}_1)\boldsymbol{\alpha}_1 &= (\boldsymbol{U}_1^\top + \boldsymbol{U}_1^\top \boldsymbol{U}_2 \boldsymbol{U}_2^\top)(\boldsymbol{x}_1 - \boldsymbol{x}_2), \\ (\boldsymbol{I} - \boldsymbol{U}_2^\top \boldsymbol{U}_1 \boldsymbol{U}_1^\top \boldsymbol{U}_2)\boldsymbol{\alpha}_2 &= (\boldsymbol{U}_2^\top + \boldsymbol{U}_2^\top \boldsymbol{U}_1 \boldsymbol{U}_1^\top)(\boldsymbol{x}_1 - \boldsymbol{x}_2), \end{aligned}$$

where $\boldsymbol{I}$ denotes the identity matrix.

In this metric, the input vector $\boldsymbol{x}$ and its transform $\boldsymbol{x} + \boldsymbol{U}\boldsymbol{\alpha}$ are equivalent. If the number of the input dimension d is much larger than the dimension of the tangent space, m, i.e., $d << m$, the algorithm works well. However, when m is larger, linear varieties may have intersection, that is distance is zero even if input vectors are far from each other, Therefore, tangent distance method is useful only if m is sufficiently smaller than d.

2.2. Mutual subspace

Mutual subspace was proposed for sequential input vectors and has been applied to face recognition, 3D object recognition and handwritten digits recognition problems [5]. Let $\boldsymbol{x}_i^1, \ldots, \boldsymbol{x}_i^t \in \mathbb{R}^d$ be sequential input vectors from the i-th pattern. This sequence is obtained, for example, from sequential image, stereo or distributed cameras, images of a rotated or fluctuated face or an object.

The pattern is evaluated as an r-dimensional eigenspace of the sequence (it is also called Karhunen-Loève subspace or eigenface space in face recognition). We denote the eigenspace by $\mathcal{X}$, and orthonormal basis of the eigenspace by $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r$, and $\boldsymbol{U} = [\boldsymbol{u}_1 \ldots \boldsymbol{u}_r] \in \mathbb{R}^{d \times r}$. The distance between $\mathcal{X}_1$ and $\mathcal{X}_2$ is measured by the angle between the spaces.

$$\cos\theta = \|\boldsymbol{U}_1 \boldsymbol{U}_1^\top \boldsymbol{U}_2 \boldsymbol{U}_2^\top\| = \|\boldsymbol{U}_1^\top \boldsymbol{U}_2 \boldsymbol{U}_2^\top \boldsymbol{U}_1\|, \tag{3}$$

where $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ denotes the matrix of orthonormal basis of $\mathcal{X}_1$ and $\mathcal{X}_2$ respectively, and $\|\cdot\|$ denotes the operator norm (it is also called the spectral norm) that is the same as the maximum eigenvalue.

Along with the tangent distance, if the dimension of the eigenspace, r is large, eigenspaces have intersection, and it results $\theta = 0$. Moreover, estimation of parameter, r is a sensitive problem.

3. Metric representation of input data

Both of the tangent distance and the mutual subspace use subspace that is generated from an input vector or input sequence. However, the selection of the dimension of the subspace is a sensitive problem, and if the dimension is too large, algorithm may not work well.

In our strategy, we define a metric tensor at the point of each input vector or an averaged sequence. Then an input pattern is represented by a pair of a vector and a matrix. We denote a point of the direct product space by $\boldsymbol{q} = (\boldsymbol{x}, \boldsymbol{Y}) \in \mathbb{R}^d \times \mathbb{R}^{d \times d}$.

In the case that input data are given as sequence of vectors, $\boldsymbol{x}$ is a mean vector, and $\boldsymbol{Y}$ is an inverse of variance-covariance matrix. If $\boldsymbol{Y}$ is singular, we may use regularization $\boldsymbol{Y}_\mu = \boldsymbol{Y} + \mu\boldsymbol{I}$, or modified variance-covariance matrix, suppose that eigenvector decomposition of $\boldsymbol{Y}$ is

$$\boldsymbol{Y} = \sum_{i=1}^{d} \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^\top. \tag{4}$$

Then modified matrix $\boldsymbol{Y}_\delta$ with a threshold δ is

$$\boldsymbol{Y}_\delta = \sum_{i=1}^{d} \max(\lambda_i, \delta) \boldsymbol{v}_i \boldsymbol{v}_i^\top. \tag{5}$$

As in the case of tangent distance, if we know priorknowledge of the local structure of the space, we can introduce metric,

$$\boldsymbol{Y} = (\boldsymbol{U}\boldsymbol{U}^\top + \mu\boldsymbol{I})^{-1} = \frac{1}{\mu}\left(\boldsymbol{I} - \frac{1}{\mu + 1}\boldsymbol{U}\boldsymbol{U}^\top\right), \tag{6}$$

where $\mu > 0$ is a regularization parameter. In this metric, an unit vector in the range of $\boldsymbol{U}$, $\boldsymbol{z} \in \mathcal{R}(\boldsymbol{U})$, $\|\boldsymbol{z}\| = 1$ is $\|\boldsymbol{z}\|_{\boldsymbol{Y}}^2 = \langle \boldsymbol{z}, \boldsymbol{Y}\boldsymbol{z} \rangle = \frac{1}{1+\mu}$, where as for an unit vector, $\boldsymbol{z} \in \mathcal{R}(\boldsymbol{U})^\perp$, $\|\boldsymbol{z}\|_{\boldsymbol{Y}}^2 = \frac{1}{\mu}$. Thus, for direction along $\boldsymbol{U}$, distance is smaller, and for direction that is orthogonal to $\boldsymbol{U}$, distance is larger.

4. Similarity measurement

We next define a similarity between two input patterns, $\boldsymbol{q}_1 = (\boldsymbol{x}_1, \boldsymbol{Y}_1)$ and $\boldsymbol{q}_2 = (\boldsymbol{x}_2, \boldsymbol{Y}_2)$. In this work, we propose two measurements named metric interpolation function (MIF) and highest valley function (HVF).

4.1. Metric interpolation function

In differential geometry, when a point $\boldsymbol{x}$ and metric at the point $\boldsymbol{x}$, $\boldsymbol{G_x}$ is given, infinitesimal distance between $\boldsymbol{x}$ and $\boldsymbol{x}+\delta\boldsymbol{x}$ is defined by $\sqrt{\langle\delta\boldsymbol{x}, \boldsymbol{G_x}\delta\boldsymbol{x}\rangle}$. We linearly interpolate matrices between two patterns. At the inner point $\boldsymbol{p}(t) = t\boldsymbol{x}_1 + (1-t)\boldsymbol{x}_2$, $(0 \leq t \leq 1)$, we define interpolated metric $\boldsymbol{M}(t)$ by

$$\boldsymbol{M}(t) = t\boldsymbol{Y}_1 + (1-t)\boldsymbol{Y}_2. \tag{7}$$

Since $\partial\boldsymbol{p}/\partial t = \boldsymbol{x}_1 - \boldsymbol{x}_2$, distance D_{MIF} is obtained by

$$D_{\mathrm{MIF}}(\boldsymbol{q}_1, \boldsymbol{q}_2) = \int_0^1 \sqrt{\langle(\boldsymbol{x}_1 - \boldsymbol{x}_2), \boldsymbol{M}(t)(\boldsymbol{x}_1 - \boldsymbol{x}_2)\rangle}\, dt. \tag{8}$$

Let

$$\begin{aligned} a &= \langle\boldsymbol{x}_1 - \boldsymbol{x}_2, (\boldsymbol{Y}_1 - \boldsymbol{Y}_2)(\boldsymbol{x}_1 - \boldsymbol{x}_2)\rangle & (9)\\ b &= \langle\boldsymbol{x}_1 - \boldsymbol{x}_2, \boldsymbol{Y}_2(\boldsymbol{x}_1 - \boldsymbol{x}_2)\rangle, & (10)\end{aligned}$$

then we have

$$D_{\mathrm{MIF}}(\boldsymbol{q}_1, \boldsymbol{q}_2) = \begin{cases} \dfrac{2}{3a}\left((a+b)^{3/2} - b^{3/2}\right) & (a \neq 0) \\ 0 & (a = 0). \end{cases} \tag{11}$$

D_{MIF} satisfies

$$\begin{aligned} D_{\mathrm{MIF}}(\boldsymbol{q}_1, \boldsymbol{q}_2) = 0 &\iff \boldsymbol{x}_1 = \boldsymbol{x}_2 & (12)\\ D_{\mathrm{MIF}}(\boldsymbol{q}_1, \boldsymbol{q}_2) &= D_{\mathrm{MIF}}(\boldsymbol{q}_2, \boldsymbol{q}_1). & (13)\end{aligned}$$

If metric $\boldsymbol{Y}$ is given by eq. (6), and rank of $\boldsymbol{U}$ is small, following expressions reduce computational complexity.

$$\begin{aligned} a &= \frac{1}{\mu(\mu+1)}(\|\boldsymbol{U}_2^\top(\boldsymbol{x}_1 - \boldsymbol{x}_2)\|^2 - \|\boldsymbol{U}_1^\top(\boldsymbol{x}_1 - \boldsymbol{x}_2)\|^2)\\ b &= \frac{1}{\mu}\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2 - \frac{1}{\mu(\mu+1)}\|\boldsymbol{U}_2^\top(\boldsymbol{x}_1 - \boldsymbol{x}_2)\|^2. \end{aligned}$$

4.2. Highest valley function

MIF utilizes two metric matrices by interpolation. However, the shortest path between two points is not always straight line. In differential geometries, the shortest path is defined by a geodesic. Since we can utilize metric tensor at only two points, it is difficult to determine its geodesic. We therefore introduce a concept of a highest valley.

Consider a quadratic function that is derived from log probability density function of the normal distribution $\mathcal{N}(\boldsymbol{x}, \boldsymbol{Y}^{-1})$ ($\boldsymbol{Y}^{-1}$ is an variance-covariance matrix),

$$q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{Y}) = -\langle\boldsymbol{z} - \boldsymbol{x}, \boldsymbol{Y}(\boldsymbol{z} - \boldsymbol{x})\rangle. \tag{14}$$

This function has maximum 0 at $\boldsymbol{z} = \boldsymbol{x}$, and its contour is ellipsoidal that is depends on metric $\boldsymbol{Y}$. For given two functions $q_1(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{Y}_1)$ and $q_2(\boldsymbol{z}|\boldsymbol{x}_2, \boldsymbol{Y}_2)$, their intersection draws quadratic curved line, surface or hyper-surface. We call this intersection the valley of q_1 and q_2.

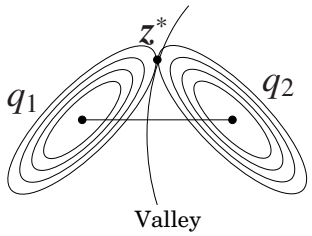

Figure 1. Example of highest valley function; Ellipsoids stand for contours of functions, q_1 and q_2

We define a similarity as the maximum value of q_1 on on the valley of q_1 and q_2. The optimization problem is given as

$$\begin{array}{ll} \max_{\boldsymbol{z}} & q_1(\boldsymbol{z}) \\ \text{subject to} & q_1(\boldsymbol{z}) = q_2(\boldsymbol{z}). \end{array} \tag{15}$$

Suppose that the optimal point is $\boldsymbol{z}^*$, then the dissimilarity is given as $D_{\mathrm{HVF}}(q_1, q_2) = q_1(\boldsymbol{z}^*)$ (or equivalent to $q_2(\boldsymbol{z}^*)$). The optimal point $\boldsymbol{z}^*$ could be far from the middle point of two centroid (Fig. 1). This represents that transformed patterns from both centroids joined at $\boldsymbol{z}^*$, and its value $q_1(\boldsymbol{z}^*) = q_2(\boldsymbol{z}^*)$ is the distance from the centroids.

It is obvious that this dissimilarity satisfies

$$\begin{aligned} D_{\mathrm{HVF}}(q_1, q_2) = 0 &\iff \boldsymbol{x}_1 = \boldsymbol{x}_2 & (16)\\ D_{\mathrm{HVF}}(q_1, q_2) &= D_{\mathrm{HVF}}(q_2, q_1). & (17)\end{aligned}$$

Semi-definite programming It is known that quadratic optimization problems on ellipsoidal constraint reduced to a semi-definite programming (SDP) [9, 11]. We rewrite the optimization problem (15),

$$\begin{array}{ll} \min_{\boldsymbol{z}} & f(\boldsymbol{z}) = \langle\boldsymbol{z}, \boldsymbol{Y}_1\boldsymbol{z}\rangle - 2\langle\boldsymbol{z}, \boldsymbol{Y}_1\boldsymbol{x}_1\rangle \\ \text{subject to} & \langle\boldsymbol{z}, \boldsymbol{Q}\boldsymbol{z}\rangle + 2\langle\boldsymbol{c}, \boldsymbol{z}\rangle = r, \end{array} \tag{18}$$

where $\boldsymbol{Q} = \boldsymbol{Y}_1 - \boldsymbol{Y}_2$, $\boldsymbol{c} = \boldsymbol{Y}_2\boldsymbol{x}_2 - \boldsymbol{Y}_1\boldsymbol{x}_1$, $r = \langle\boldsymbol{x}_1, \boldsymbol{Y}_1\boldsymbol{x}_1\rangle - \langle\boldsymbol{x}_2, \boldsymbol{Y}_2\boldsymbol{x}_2\rangle$. Let ξ be a Lagrange multiplier. From Karush-Kuhn-Tucker (KKT) conditions, the optimal point $\boldsymbol{z}^*, \xi^*$ satisfies

$$\boldsymbol{Y}_1\boldsymbol{z}^* - \boldsymbol{Y}_1\boldsymbol{x}_1 + \xi^*(\boldsymbol{Q}\boldsymbol{z}^* + \boldsymbol{c}) = \boldsymbol{0}. \tag{19}$$

From the second order of KKT condition, $\boldsymbol{Y}_1 + \xi^*\boldsymbol{Q} \succeq 0$, where $\succeq$ denotes the semi-positive definite [1]. From eq. (19), we have

$$\begin{aligned} \langle\boldsymbol{z}^*, \boldsymbol{Y}_1\boldsymbol{z}^*\rangle - \langle\boldsymbol{z}^*, \boldsymbol{Y}_1\boldsymbol{x}_1\rangle &= -\xi^*(\langle\boldsymbol{z}^*, \boldsymbol{Q}\boldsymbol{z}^*\rangle + \langle\boldsymbol{c}, \boldsymbol{z}^*\rangle)\\ f(\boldsymbol{z}^*) + \langle\boldsymbol{z}^*, \boldsymbol{Y}_1\boldsymbol{x}_1\rangle &= -\xi^* r - \xi^*\langle\boldsymbol{c}, \boldsymbol{z}^*\rangle \end{aligned}$$

Let s be a lower bound of $f(\boldsymbol{z}^*)$

$$\begin{aligned} f(\boldsymbol{z}^*) &= -\xi^* r - \xi^* \langle \boldsymbol{c}, \boldsymbol{z}^* \rangle - \langle \boldsymbol{z}^*, \boldsymbol{Y}_1 \boldsymbol{x}_1 \rangle \\ &\geq s. \end{aligned}$$

Suppose that $\boldsymbol{Y}_1 + \xi^* \boldsymbol{Q}$ is not singular. From $\boldsymbol{z}^* = (\boldsymbol{Y}_1 + \xi^* \boldsymbol{Q})^{-1}(\boldsymbol{Y}_1 \boldsymbol{x}_1 - \xi^* c)$,

$$\begin{aligned} &-\xi^* r - s - \\ &\langle (\boldsymbol{Y}_1 \boldsymbol{x}_1 - \xi^* \boldsymbol{c}), (\boldsymbol{Y}_1 + \xi^* \boldsymbol{Q})^{-1} (\boldsymbol{Y}_1 \boldsymbol{x}_1 - \xi^* \boldsymbol{c}) \rangle \geq 0. \end{aligned}$$

This is called the Schur complement and the inequality is satisfied if and only if

$$\begin{bmatrix} \boldsymbol{Y}_1 + \xi^* \boldsymbol{Q} & \boldsymbol{Y}_1 \boldsymbol{x}_1 - \xi^* \boldsymbol{c} \\ (\boldsymbol{Y}_1 \boldsymbol{x}_1 - \xi^* \boldsymbol{c})^\top & -\xi^* r - s \end{bmatrix} \succeq 0 \tag{20}$$

is satisfied [3]. Then the problem is reduced to

$$\begin{array}{ll} \max_{\xi^*} & s \\ \text{subject to} & \text{(20) is satisfied.} \end{array} \tag{21}$$

This is a kind of SDP that is solved in polynomial order complexity. If $\boldsymbol{Y}_1 + \xi^* \boldsymbol{Q}$ has zero eigenvalues, minimum norm solution $\boldsymbol{z}^* = (\boldsymbol{Y}_1 + \xi^* \boldsymbol{Q})^\dagger (\boldsymbol{Y}_1 \boldsymbol{x}_1 - \xi^* \boldsymbol{c})$ is used.

Linear search method Although SDP is solved in polynomial time, its computational complexity is high. We therefore introduce an linear search based optimization method for the problem. The Lagrange function is given by

$$L(\boldsymbol{z}, \xi) = \langle \boldsymbol{z}, (\boldsymbol{Y}_1 + \xi \boldsymbol{Q}) \boldsymbol{z} \rangle - 2 \langle \boldsymbol{z}, \boldsymbol{Y}_1 \boldsymbol{x}_1 - \xi \boldsymbol{c} \rangle - \xi r. \tag{22}$$

Since the constraint of the problem (15) is equivalent to the constraint, $q_1(\boldsymbol{z}) \leq q_2(\boldsymbol{z})$, the dual problem is

$$\begin{array}{ll} \max_{\xi} & L(\xi) \\ \text{subject to} & \boldsymbol{z} = (\boldsymbol{Y}_1 + \xi \boldsymbol{Q})^{-1} (\boldsymbol{Y}_1 \boldsymbol{x}_1 - \xi \boldsymbol{c}), \\ & \xi \leq 0. \end{array} \tag{23}$$

Since $\boldsymbol{Y}_1$ is positive definite matrix, minimum norm solution provides the optimal solution if $\boldsymbol{Y}_1 + \xi \boldsymbol{Q}$ is singular, The dual problem (23) has only one variable, we can solve the problem by using 1-dimensional linear search. We provide the algorithm;

1. Set initial value for $(\xi_1, \xi_2, \xi_3) = (-1, -0.5, 0)$ because $\boldsymbol{Y}_1 + \xi \boldsymbol{Q}$ is not singular when $\xi \in [-1, 0]$.
2. Obtain $L(\xi_1)$, $L(\xi_2)$, $L(\xi_3)$.
3. Iterate following step until it converges;
   (a) Find expected maximum point ξ_4 using 1-dimensional search such as the golden section search [4] or parabolic interpolation.
   (b) If $\xi > 0$, set $\xi \leftarrow 0$.
   (c) Exchange the point

$$\begin{aligned} k &= \text{argmin}_{i=1,2,3} L(\xi_i) \\ \xi_k &\leftarrow \xi_4, \; L(\xi_k) \leftarrow L(\xi_4) \end{aligned}$$

   (d) Go to (a).

If the difference $|\xi_k - \xi_4| < \epsilon$ or $|L(\xi_k) - L(\xi_4)| < \epsilon$ the algorithm is stopped, where ϵ is small constant, say $\epsilon = 10^{-6}$.

5. Experiment

We used USPS handwritten digits dataset to demonstrate our algorithm. It consists of 7,291 samples for training and 2,007 samples for testing. All samples are given by 16x16 pixel gray-scale image. Seven tangent vectors in [7] were made from each samples, and its metric tensor was obtained from eq. (6). We employed the nearest neighbor rule under several distance measurement. We compared the proposed methods over five conventional methods;

1. k-nearest neighbor (kNN) with Euclidean distance.
2. Mutual subspace method (Msub) [5].
3. One-side tangent distance (TD1) [7].
4. Two-side tangent distance (TD2) [7].
5. Mahalanobis metric.

Mahalanobis metric was obtained with regularization for each classes from training samples. Table 1 summarized the results of the recognition test. In [7], the error rate of TD2 for USPS dataset is 2.5%. This is due to the parameter to obtain tangent vector is not the same. From table 1, HVF shows the lowest error rate. MIF is higher error rate then TD2. However, if we use the expanded forms of a and b, the calculation cost will much smaller than TD2.

Table 2 shows the comparison of calculation times. We used a computer with AMD Phenom(TM) 9550 Quad-core processor 2.2GHz, and 4 Gbyte RAM. The implementation is done by GNU octave, and SDP is solved by SDPA [12]. The calculation times are for one test sample, i.e., 7,291 times evaluations of distance measurement. From Table 2, we can see that MIF is much faster than TD2. Although our linear search algorithm is much faster than SDP, the linear search algorithm costs a lot of computational complexity. As described in [6], we can reduce calculation costs using ordinary kNN rule as a pre-processing.

In digit or letter classification problems, since spaces of the tangent distance are predefined [6], TD1 and TD2 have

Table 1. Experimental result for handwritten digits recognition problem

Method	Parameter	Error rate [%]
MIF	$\mu = 0.1$	3.84
HVF	$\mu = 10^{-1.6}$	3.49
kNN	$k = 1$	5.53
Msub	$r = 7$	7.87
TD1	–	4.73
TD2	–	3.59
Mahalanobis	$\mu = 10$	9.82

Table 2. Comparison of run-times

Method	run-time [s]
MIF	6.4
HVF (linear search)	390
HVF (SDP)	19900
TD1	2.5
TD2	17

no parameter. However, in general problems, we have to determine their tangent subspaces, and they have parameters such as the number of dimension.

Both of proposed methods have a parameter μ, this is not from their definition but the definition of metric. In this example, we used the metric described in Section 3, however, the proposed methods do not limit metrics.

6. Conclusion

We systematize the framework that describes an input pattern as a d-dimensional vector $\boldsymbol{x}$ and $d \times d$ matrix $\boldsymbol{Y}$ that is metric at the point $\boldsymbol{x}$. Then, we proposed two new measurements of similarities into this framework, one is the metric interpolation function (MIF) and the other is the highest valley function (HVF). HVF is reduced to the semidefinite programming (SDP) that has high computational complexity. We also proposed the algorithm using linear search algorithm for HVF. From the experimental results, HVF showed the lowest error rate. MIF was not higher accuracy than TD2, but MIF was much faster than TD2. However, our methods are useful for given metric, and estimation of the optimal metric is also an important problem.

Acknowledgement

This research was supported by a Grant-in-Aid for Young Scientists (B), No. 19700153, from Japan Society for the Promotion of Science (JSPS).

References

[1] J. M. Borwein and A. S. Lewis. *Convex analysis and nonlinear optimization*. Springer, 2000.

[2] A. Globerson and S. Roweis. Metric learning by collapsing classes. In *Advances in Neural Information Processing Systems 18*, pages 451–458. MIT Press, 2006.

[3] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge university press, 1985.

[4] J. Kiefer. Sequential minimax search for a maximum. *Proc. of the American Mathematical Society*, 4:502–506, 1953.

[5] K. Maeda and S. Watanabe. A pattern matching method with local structure. *IEICE trans. on information and systems (D) (in Japanese)*, J68-D(3):345–352, 1985.

[6] P. Y. Simard, Y. A. LeCun, and J. S. Denker. Efficient pattern recognition using a new transformation distance. *Advances in Neural Information Processing Systems*, 5:50–58, 1992.

[7] P. Y. Simard, Y. A. LeCun, J. S. Denker, and B. Victorri. Transformation invariance in pattern recognition – tangent distance and tangent propagation. *Lecture Notes in Computer Science*, 1524:239–274, 1998.

[8] M. Sugiyama. Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis. *Journal of machine learning research*, 8:1027–1061, 2007.

[9] H. Tuy. *Convex analysis and global optimization*. Kluwer academic publishers, 1998.

[10] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems 18*, pages 1473–1480. MIT Press, 2006.

[11] Y. Yajima. Non-convex quadratic optimization problem and combinatorial optimization. *Operations research as a management science research (in Japanese)*, 44(5):237–242, 1999.

[12] M. Yamashita, K. Fujisawa, and M. Kojima. Implementation and evaluation of SDPA 6.0 (semidefinite programming algorithm 6.0). *Optimization Methods and Software*, pages 491–505, 2003.

[13] L. Yang and R. Jin. An efficient algorithm for local distance metric learning. In *in Proceedings of AAAI*, pages 543–548, 2006.