Two-Stage Metric Learning

Jun Wang JUN.WANG@UNIGE.CH
Department of Computer Science, University of Geneva, Switzerland

Ke Sun KE.SUN@UNIGE.CH
Department of Computer Science, University of Geneva, Switzerland

Fei Sha FEISHA@USC.EDU
Department of Computer Science, University of Southern California, Los Angeles, CA, USA

Stephane Marchand-Maillet STEPHANE.MARCHAND-MAILLET@UNIGE.CH
Department of Computer Science, University of Geneva, Switzerland

Alexandros Kalousis ALEXANDROS.KALOUSIS@HESGE.CH
Department of Business Informatics,University of Applied Sciences,Western Switzerland,
Department of Computer Science, University of Geneva, Switzerland

Abstract

In this paper, we present a novel two-stage metric learning algorithm. We first map each learning instance to a probability distribution by computing its similarities to a set of fixed anchor points. Then, we define the distance in the input data space as the Fisher information distance on the associated statistical manifold. This induces in the input data space a new family of distance metric with unique properties. Unlike kernelized metric learning, we do not require the similarity measure to be positive semi-definite. Moreover, it can also be interpreted as a local metric learning algorithm with well defined distance approximation. We evaluate its performance on a number of datasets. It outperforms significantly other metric learning methods and SVM.

1. Introduction

Distance measures play a crucial role in many machine learning tasks and algorithms. Standard distance metrics, e.g. Euclidean, cannot address in a satisfactory manner the multitude of learning problems, a fact that led to the development of metric learning methods which learn problem-specific distance measure directly from the data (Weinberger & Saul, 2009; Wang et al., 2012; Jain et al., 2010).

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

Over the last years various metric learning algorithms have been shown to perform well in different learning problems, however, each comes with its own set of limitations.

Learning the distance metric with one global linear transformation is called single metric learning (Weinberger & Saul, 2009; Davis et al., 2007). In this approach the distance computation is equivalent to applying on the learning instances a learned linear transformation followed by a standard distance metric computation in the projected space. Since the discriminatory power of the input features might vary locally, this approach is often not flexible enough to fit well the distance in different regions.

Local metric learning addresses this limitation by learning in each neighborhood one local metric (Noh et al., 2009; Wang et al., 2012). When the local metrics vary smoothly in the feature space, learning local metrics is equivalent to learning the Riemannian metric on the data manifold (Hauberg et al., 2012). The main challenge here is that the geodesic distance endowed by the Riemannian metric is often computationally very expensive. In practice, it is approximated by assuming that the geodesic curves are formed by straight lines and the local metric does not change along these lines (Noh et al., 2009; Wang et al., 2012). Unfortunately, the approximation does not satisfy the symmetric property and therefore the result is a non-metric distance.

Kernelized Metric Learning (KML) achieves flexibility in a different way (Jain et al., 2010; Wang et al., 2011). In KML learning instances are first mapped into the Reproducing-Kernel Hilbert Space (RKHS) by a kernel

function and then a global Mahalanobis metric is learned in the RKHS space. By defining the distance in the input feature space as the Mahalanobis distance in the RKHS space, KML is equivalent to learning a flexible non-linear distance in the input space. However, its main limitation is that the kernel matrix induced by the kernel function must be Positive Semi-Definite (PSD). Although Non-PSD kernel could be transformed into PSD kernel (Chen & Ye, 2008; Ying et al., 2009), the new PSD kernel nevertheless cannot keep all original similarity information.

In this paper, we propose a novel two-stage metric learning algorithm, Similarity-Based Fisher Information Metric Learning (SBFIML). It first maps instances from the data manifold into finite discrete distributions by computing their similarities to a number of predefined anchor points in the data space. Then, the Fisher information distance on the statistical manifold is used as the distance in the input feature space. This induces a new family of Riemannian distance metric in the input data space with two important properties. First, the new Riemannian metric is robust to density variation in the original data space. Without such robustness, an objective function can be easily biased towards data regions the density of which is low and thus dominates learning of the objective function. Second, the new Riemannian metric has largest distance discrimination on the manifold of anchor points and no distance in the directions being orthogonal to the manifold. So, the effect of locally irrelevant dimensions of anchor points is removed. To the best of our knowledge, this is the first metric learning algorithm that has these two important properties.

SBFIML is flexible and general; it can be applied to different types of data spaces with various non-negative similarity functions. Comparing to KML, SBFIML does not require the similarity measure to form a PSD matrix. Moreover, SBFIML can be interpreted as a local metric learning algorithm. Compared to the previous local metric learning algorithms which produce a non-metric distance (Noh et al., 2009; Wang et al., 2012), the distance approximation in SBFIML is a well defined distance function with a closed form expression. We evaluate SBFIML on a number of datasets. The experimental results show that it outperforms in a statistically significant manner both metric learning methods and SVM.

2. Preliminaries

We are given a number of learning instances $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, where each instance $\boldsymbol{x}_i^T \in \mathcal{X}$ is a d-dimensional vector, and a vector of associated class labels $\boldsymbol{y} = (y_1, \ldots, y_n)^T$, $y_i \in \{1, \ldots, c\}$. We assume that the input feature space $\mathcal{X}$ is a smooth manifold. Different learning problems can have very different types of data manifolds with possibly different dimensionality. The most commonly used manifold in metric learning is the Euclidean space $\mathbb{R}^d$ (Weinberger & Saul, 2009). The probability simplex space $\mathcal{P}^{d-1}$ has also been explored (Lebanon, 2006; Cuturi & Avis, 2011; Kedem et al., 2012).

We propose a general two-stage metric learning algorithm which can learn a flexible distance in different types of $\mathcal{X}$ data manifolds, e.g. Euclidean, probability simplex, hypersphere, etc. Concretely, we first map instances from $\mathcal{X}$ onto the statistical manifold $\mathcal{S}$ through a similarity-based differential map, which computes their non-negative similarities to a number of predefined anchor points. Then we define the Fisher information distance as the distance on $\mathcal{X}$. We have chosen to do so, since this induces a new family of Riemannian distance metric which enjoys interesting properties: 1) The new Riemannian metric is robust to density variations in the original data space, which can be produced for example by different intrinsic variabilities of the learning instances in the different categories. Distance learning over this new metric is hence robust to density variation. 2) The new Riemannian distance metric has largest distance discrimination on the manifold of the anchor points and has no distance in the directions being orthogonal to that manifold. So, the new distance metric can remove the effect of locally irrelevant dimensions of the anchor point manifold, see Figure 1 for more detials. In the remainder of this section, we will briefly introduce the necessary terminology and concepts. More details can be found in the monographs (Lee, 2002; Amari & Nagaoka, 2007).

Statistical Manifold. We denote by $\mathcal{M}^n$ a n-dimensional smooth manifold. For each point p on $\mathcal{M}^n$, there exists at least one smooth coordinate chart $(\mathcal{U}, \varphi)$ which defines a coordinate system to points on $\mathcal{U}$, where $\mathcal{U}$ is an open subset of $\mathcal{M}^n$ containing p and $\varphi : \mathcal{U} \longrightarrow \Theta$ is a smooth coordinate map $\varphi(p) = \boldsymbol{\theta} \in \Theta \subset \mathbb{R}^n$. $\boldsymbol{\theta}$ is the coordinate of p defined by φ.

A statistical manifold is a smooth manifold whose points are probability distributions. Given a n-dimensional statistical manifold $\mathcal{S}^n$, we denote by $p(\boldsymbol{\xi}|\boldsymbol{\theta})$ a probability distribution in $\mathcal{S}^n$, where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n) \in \Theta \subset \mathbb{R}^n$ is the coordinate of $p(\boldsymbol{\xi}|\boldsymbol{\theta})$ under some coordinate map φ and $\boldsymbol{\xi}$ is the random variable of the $p(\boldsymbol{\xi}|\boldsymbol{\theta})$ distribution taking values from some set Ξ. Note that, all the probability distributions in $\mathcal{S}^n$ share the same set Ξ.

In this paper, we are particularly interested in the n-dimensional statistical manifold $\mathcal{P}^n$, whose points are finite discrete distributions, denoted by

$$\mathcal{P}^n = \{p(\xi|\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)) : \sum_{i=1}^{n} \theta_i < 1, \forall i, \theta_i > 0\} \quad (1)$$

where ξ is the discrete random variable taking values in the set $\Xi = \{1, \ldots, n+1\}$ and $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^n$ is called the m-affine coordinate (Amari & Nagaoka, 2007). The

probability mass of $p(\xi|\boldsymbol{\theta})$ is $p(\xi = i) = \theta_i$ if $i \neq n+1$, otherwise $p(\xi = n+1) = 1 - \sum_{k=1}^{n} \theta_k$.

Fisher Information Metric. The Fisher information metric is a Riemannian metric defined on statistical manifolds and endows a distance between probability distributions (Radhakrishna Rao, 1945). The explicit form of the Fisher information metric at $p(\boldsymbol{\xi}|\boldsymbol{\theta})$ is a $n \times n$ positive definite symmetric matrix $\mathbf{G}_{FIM}(\boldsymbol{\theta})$, the (i,j) element of which is defined by:

$$\mathbf{G}^{ij}_{FIM}(\boldsymbol{\theta}) = \int_{\Xi} \frac{\partial \log p(\boldsymbol{\xi}|\boldsymbol{\theta})}{\partial \theta_i} \frac{\partial \log p(\boldsymbol{\xi}|\boldsymbol{\theta})}{\partial \theta_j} p(\boldsymbol{\xi}|\boldsymbol{\theta}) d\boldsymbol{\xi} \quad (2)$$

where the above integral is replaced with a sum if Ξ is discrete. The following lemma gives the explicit form of the Fisher information metric on $\mathcal{P}^n$.

Lemma 1. *On the statistical manifold $\mathcal{P}^n$, the Fisher information metric $\mathbf{G}_{FIM}(\boldsymbol{\theta})$ at $p(\xi|\boldsymbol{\theta})$ with coordinate $\boldsymbol{\theta}$ is*

$$\mathbf{G}^{ij}_{FIM}(\boldsymbol{\theta}) = \frac{1}{\theta_i}\delta_{ij} + \frac{1}{1 - \sum_{k=1}^{n} \theta_k}, \forall i,j \in \{1,\ldots,n\} \quad (3)$$

where $\delta_{ij} = 1$ if $i = j$, otherwise $\delta_{ij} = 0$.

Properties of Fisher Information Metric. The Fisher information metric enjoys a number of interesting properties. First, the Fisher information metric is the unique Riemannian metric induced by all f-divergence measures, such as the Kullback-Leibler (KL) divergence and the χ^2 divergence (Amari & Cichocki, 2010). All these divergences converge to the Fisher information distance as the two probability distributions are approaching each other. Another important property of the Fisher information metric from a metric learning perspective is that the distance it endows can be approximated by the Hellinger distance, the cosine distance and all f-divergence measures (Kass & Vos, 2011). More importantly, when $\mathcal{S}^n$ is the statistical manifold of finite discrete distributions, e.g. $\mathcal{P}^n$, the cosine distance is exactly equivalent to the Fisher information distance (Lebanon, 2006; Lee et al., 2007).

Pullback Metric. Let $\mathcal{M}^n$ and $\mathcal{N}^m$ be two smooth manifolds and $\mathcal{T}_p\mathcal{M}^n$ be the tangent space of $\mathcal{M}^n$ at $p \in \mathcal{M}^n$. Given a differential map $f : \mathcal{M}^n \longrightarrow \mathcal{N}^m$ and a Riemannian metric $\mathbf{G}$ on $\mathcal{N}^m$, the differential map f induces a pullback metric $\mathbf{G}^*$ at each point p on $\mathcal{M}^n$ defined by:

$$\langle \boldsymbol{v}_1, \boldsymbol{v}_2 \rangle_{\mathbf{G}^*(p)} = \langle D_p f(\boldsymbol{v}_1), D_p f(\boldsymbol{v}_2) \rangle_{\mathbf{G}(f(p))} \quad (4)$$

where $D_p f : \mathcal{T}_p\mathcal{M}^n \longrightarrow \mathcal{T}_{f(p)}\mathcal{N}^m$ is the differential of f at point $p \in \mathcal{M}^n$, which maps tangent vectors $\boldsymbol{v} \in \mathcal{T}_p\mathcal{M}^n$ to tangent vectors $D_p f(\boldsymbol{v}) \in \mathcal{T}_{f(p)}\mathcal{N}^m$.

Given the coordinate systems Θ and Γ of $\mathcal{U} \subset \mathcal{M}^n$ and $\mathcal{U}' \subset \mathcal{N}^m$ respectively, defined by some smooth coordinate maps $\varphi_{\mathcal{U}}$ and $\varphi_{\mathcal{U}'}$ respectively, then the explicit form of the pullback metric at point $p \in \mathcal{U} \subset \mathcal{M}^n$ with coordinate $\boldsymbol{\theta} = \varphi_{\mathcal{U}}(p)$ is:

$$\mathbf{G}^*(\boldsymbol{\theta}) = \mathbf{J}^T \mathbf{G}(\boldsymbol{\gamma}) \mathbf{J} \quad (5)$$

where $\boldsymbol{\gamma} = \varphi_{\mathcal{U}'}(f(p))$ is the coordinate of the $f(p) \in \mathcal{U}' \subset \mathcal{N}^m$ and $\mathbf{J}$ is the Jacobian matrix of the function $\varphi_{\mathcal{U}'} \circ f \circ \varphi_{\mathcal{U}}^{-1} : \Theta \longrightarrow \Gamma$ at point $\boldsymbol{\theta}$. Since $\mathbf{G}$ is a Riemannian metric, the pullback metric $\mathbf{G}^*$ is in general at least a PSD metric.

The following lemma gives the relation between the geodesic distances on $\mathcal{M}^n$ and $\mathcal{N}^m$.

Lemma 2. *Let $\mathbf{G}^*$ be the pullback metric of a Riemannian metric $\mathbf{G}$ induced by a differential map $f : \mathcal{M}^n \longrightarrow \mathcal{N}^m$, $d_{\mathbf{G}^*}(p', p)$ be the geodesic distance on $\mathcal{M}^n$ endowed by $\mathbf{G}^*$ and $d_{\mathbf{G}}(f(p'), f(p))$ the geodesic distance on $\mathcal{N}^m$ endowed by $\mathbf{G}$, then, it holds $\lim_{p' \to p} \frac{d_{\mathbf{G}}(f(p'), f(p))}{d_{\mathbf{G}^*}(p', p)} = 1$*

The proof of Lemma 2 is provided in the appendix. In addition to approximating $d_{\mathbf{G}^*}(p', p)$ directly on $\mathcal{M}^n$ by assuming that the geodesic curve is formed by straight lines as previous local metric learning algorithms do (Noh et al., 2009; Wang et al., 2012), Lemma 2 allows us to also approximate it with $d_{\mathbf{G}}(f(p'), f(p))$ on $\mathcal{N}^m$. Note that, both approximations have the same asymptotic convergence result.

3. Similarity-Based Fisher Information Metric Learning

We will now present our two-stage metric learning algorithm, SBFIML. In the following, we will first present how to define the similarity-based differential map $f : \mathcal{X} \longrightarrow \mathcal{P}$ and then how to learn the Fisher information distance.

3.1. Similarity-Based Differential Map

Given a number of anchor points $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$, $\boldsymbol{z}_i \in \mathcal{X}$, we denote by $s = (s_1, \ldots, s_n) : \mathcal{X} \longrightarrow \mathbb{R}^{+n}$ the differentiable similarity function. Each $s_k : \mathcal{X} \longrightarrow \mathbb{R}^+$ component is a differentiable function the output of which is a non-negative similarity between some input instance $\boldsymbol{x}_i$ and the anchor point $\boldsymbol{z}_k$. Based on the similarity function s we define the similarity-based differential map f as:

$$\begin{aligned} f(\boldsymbol{x}_i) &= p(\xi|(\frac{s_1(\boldsymbol{x}_i)}{\sum_{k=1}^{n} s_k(\boldsymbol{x}_i)}, \ldots, \frac{s_{n-1}(\boldsymbol{x}_i)}{\sum_{k=1}^{n} s_k(\boldsymbol{x}_i)})) \quad (6) \\ &= (\bar{s}_1(\boldsymbol{x}_i), \ldots, \bar{s}_{n-1}(\boldsymbol{x}_i)) \end{aligned}$$

where $f(\boldsymbol{x}_i)$ is a finite discrete distribution on manifold $\mathcal{P}^{n-1}$. From now on, for simplicity, we will denote $f(\boldsymbol{x}_i)$ by $p^i(\xi)$. The probability mass of the kth outcome is given by: $p^i(\xi = k) = \bar{s}_k(\boldsymbol{x}_i) = \frac{s_k(\boldsymbol{x}_i)}{\sum_{k=1}^{n} s_k(\boldsymbol{x}_i)}$. In order for f to be a valid differential map, the similarity function s must satisfy $\sum_k s_k(\boldsymbol{x}_i) > 0$, $\forall \boldsymbol{x}_i \in \mathcal{X}$. This family of differential maps is very general and can be applied to any $\mathcal{X}$ space

where a non-negative differentiable similarity function s can be defined. The finite discrete distribution representation, $p^i(\xi)$, of learning instance, $\boldsymbol{x}_i$, can be intuitively seen as an encoding of its neighborhood structure defined by the similarity function s. Note that, the idea of mapping instances onto the statistical manifold $\mathcal{P}$ has been previously studied in manifold learning, e.g. SNE (Hinton & Roweis, 2002) and t-SNE (Van der Maaten & Hinton, 2008).

Akin to the appropriate choice of the kernel function in a kernel-based method, the choice of an appropriate similarity function s is also crucial for SBFIML. In principle, an appropriate similarity function s should be a good match for the geometrical structure of the $\mathcal{X}$ data manifold. For example, for data lying on the probability simplex space, i.e. $\mathcal{X} = \mathcal{P}^{d-1}$, the similarity functions defined either on $\mathbb{R}^d$ or on $\mathcal{P}^{d-1}$ can be used. However, the similarity function on $\mathcal{P}^{d-1}$ is more appropriate, because it exploits the geometrical structure of $\mathcal{P}^{d-1}$, which, in contrast, is ignored by the similarity function on $\mathbb{R}^d$ (Kedem et al., 2012).

The set of anchor points $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ can be defined in various ways. Ideally, anchor points should be similar to the given learning instances $\boldsymbol{x}_i$, i.e. anchor points follow the same distribution as that of learning instances. Empirically, we can use directly training instances or cluster centers, the latter established by clustering algorithms. Similar to the current practice in kernel methods we will use in SBFIML as anchors points all the training instances.

Similarity Functions on $\mathbb{R}^d$. We can define the similarity on $\mathbb{R}^d$ in various ways. In this paper we will investigate two types of differentiable similarity functions. The first one is based on the Gaussian function, defined as:

$$s_k(\boldsymbol{x}_i) = exp(-\frac{\|\boldsymbol{x}_i - \boldsymbol{z}_k\|_2^2}{\sigma_k}) \tag{7}$$

where $\|\cdot\|_2$ is the L_2 norm. σ_k controls the size of the neighborhood of the anchor point $\boldsymbol{z}_k$, with large values producing large neighborhoods. Note that the different σ_ks could be set to different values; if all of them are equal, this similarity function is exactly the Gaussian kernel. The second type of similarity function that we will look at is:

$$s_k(\boldsymbol{x}_i) = 1 - \frac{1}{\pi}\arccos(\frac{\boldsymbol{x}_i^T \boldsymbol{z}_k}{\|\boldsymbol{x}_i\|_2 \cdot \|\boldsymbol{z}_k\|_2}) \tag{8}$$

which measures the normalized angular similarity between $\boldsymbol{x}_i$ and $\boldsymbol{z}_k$. This similarity function can be explained as we first projecting all points from $\mathbb{R}^d$ to the hypersphere and then applying the angular similarity to points on a hypersphere. As a result, this similarity function is useful for data which approximately lie on a hypersphere. Note that this similarity function is also a valid kernel function (Honeine & Richard, 2010).

One might say we can also achieve nonlinearity by mapping instances into the proximity space $\mathcal{Q}$ using the following similarity-based map $g : \mathcal{X} \longrightarrow \mathcal{Q}$:

$$g(\boldsymbol{x}) = (s_1(\boldsymbol{x}), \ldots, s_n(\boldsymbol{x})) \tag{9}$$

We now compare our similarity-based map f, equation 6 against the similarity-based map g, equation 9, in two aspects, namely representation robustness and pullback metric analysis.

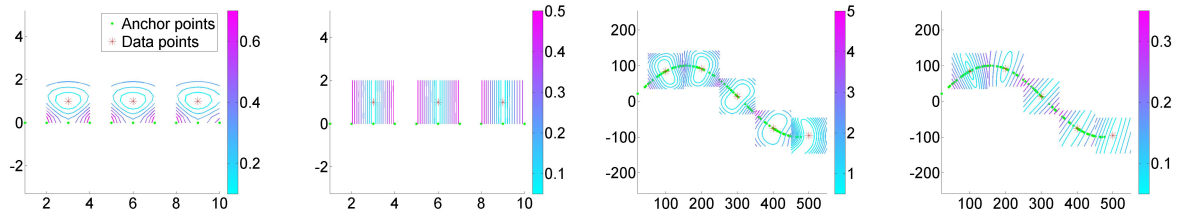
Representation Robustness. Compared to the representation induced by the similarity-based map g, equation 9, our representation induced by the similarity-based map f, equation 6, is more robust to density variations in original data space, i.e. the density of the learning instances varies significantly between different regions. This can be explained by the fact that the finite discrete distribution is essentially a representation of the neighborhood structure of a learning instance normalized by a "scaling" factor, the sum of similarities of the learning instance to the anchor points. Hence the distance implied by the finite discrete distribution representation is less sensitive to the density variations of the different data regions. This is an important property. Without such robustness, an objective function based on raw distances can be easily biased towards data regions the density of which is low and thus dominates learning of the objective function. One example of this kind of objective is that of LMNN (Weinberger & Saul, 2009), which we will also use later in SBFIML to learn the Fisher information distance.

Pullback Metric Analysis. We also show how the two approaches differ by comparing the pullback metrics induced by the two similarity-based maps f and g. In doing so, we first need to specify the Riemannian metrics $\mathbf{G}_{\mathcal{Q}}$ in the proximity space $\mathcal{Q}$ and $\mathbf{G}_{\mathcal{P}}$ on the statistical manifold $\mathcal{P}^{n-1}$. Following the work of similarity-based learning (Chen et al., 2009), we use the Euclidean metric as the $\mathbf{G}_{\mathcal{Q}}$ in the proximity space $\mathcal{Q}$. On the statistical manifold $\mathcal{P}^{n-1}$ we use the Fisher information metric $\mathbf{G}_{FIM}$ defined in equation 3 as $\mathbf{G}_{\mathcal{P}}$. To simplify our analysis, we assume $\mathcal{X} = \mathbb{R}^d$. However, note that this analysis can be generalized to other manifolds, e.g. $\mathcal{P}^{d-1}$. We use the standard Cartesian coordinate system for points in $\mathbb{R}^d$ and $\mathcal{Q}$ and use m-affine coordinate system, equation 1, for points on $\mathcal{P}^{n-1}$.

The pullback metric induced by these two differential maps are given in the following lemma.

Lemma 3. *In $\mathbb{R}^d$, at $\boldsymbol{x}$ with Cartesian coordinate, the form of the pullback metric $\mathbf{G}^*_{\mathcal{Q}}(\boldsymbol{x})$ of the Euclidean metric induced by the differential map g of equation 9 is:*

$$\mathbf{G}^*_{\mathcal{Q}}(\boldsymbol{x}) = \nabla g(\boldsymbol{x}) \nabla g(\boldsymbol{x})^T = \sum_{i=1}^{n} \nabla s_i(\boldsymbol{x}) \nabla s_i(\boldsymbol{x})^T \tag{10}$$



(a) $\mathbf{G}^*_{\mathcal{Q}}(\boldsymbol{x})$ (b) $\mathbf{G}^*_{\mathcal{P}}(\boldsymbol{x})$ (c) $\mathbf{G}^*_{\mathcal{Q}}(\boldsymbol{x})$ (d) $\mathbf{G}^*_{\mathcal{P}}(\boldsymbol{x})$

Figure 1. The visulization of equi-distance curves of pullback metrics $\mathbf{G}^*_{\mathcal{Q}}(\boldsymbol{x})$ and $\mathbf{G}^*_{\mathcal{P}}(\boldsymbol{x})$.

where the vector $\nabla s_i(\boldsymbol{x})$ of size $d \times 1$ is the differential of ith similarity function $s_i(\boldsymbol{x})$. The form of the pullback metric $\mathbf{G}^*_{\mathcal{P}}(\boldsymbol{x})$ of the Fisher information metric induced by the differential map f of equation 6 is:

$$\mathbf{G}^*_{\mathcal{P}}(\boldsymbol{x}) = \sum_{i=1}^{n} \frac{1}{\bar{s}_i(\boldsymbol{x})} (\nabla \bar{s}_i(\boldsymbol{x}) \nabla \bar{s}_i(\boldsymbol{x})^T) \quad (11)$$

where $\nabla \bar{s}_i(\boldsymbol{x}) = \bar{s}_i(\boldsymbol{x}) \left(\nabla \log(s_i(\boldsymbol{x})) - E\left(\nabla \log(s_i(\boldsymbol{x}))\right)\right)$ and the expectation of $\nabla \log(s_i(\boldsymbol{x}))$ is $E(\nabla \log(s_i(\boldsymbol{x}))) = \sum_{k=1}^{n} \bar{s}_k(\boldsymbol{x}) \nabla \log(s_i(\boldsymbol{x}))$.

Gaussian Similarity Function. The form of pullback metrics $\mathbf{G}^*_{\mathcal{Q}}(\boldsymbol{x})$ and $\mathbf{G}^*_{\mathcal{P}}(\boldsymbol{x})$ depends on the explicit form of the similarity function $s_i(\boldsymbol{x})$. We now study their differences using the Gaussian similarity function with kernel width σ, equation 7. We first show the difference between $\mathbf{G}^*_{\mathcal{Q}}(\boldsymbol{x})$ and $\mathbf{G}^*_{\mathcal{P}}(\boldsymbol{x})$ by comparing their m largest eigenvectors, the directions in which metrics have the largest distance discrimination.

The m largest eigenvectors $\mathbf{U}_{\mathcal{Q}}(\boldsymbol{x})$ of $\mathbf{G}^*_{\mathcal{Q}}(\boldsymbol{x})$ are:

$$\begin{aligned} \mathbf{U}_{\mathcal{Q}}(\boldsymbol{x}) &= \arg\max_{\mathbf{U}^T\mathbf{U}=\mathbf{I}} tr(\mathbf{U}^T \mathbf{G}^*_{\mathcal{Q}}(\boldsymbol{x}) \mathbf{U}) \quad (12) \\ &= \arg\max_{\mathbf{U}^T\mathbf{U}=\mathbf{I}} \sum_{k=1}^{m} \sum_{i=1}^{n} \frac{4}{\sigma^2} (\boldsymbol{u}_k^T s_i(\boldsymbol{x})(\boldsymbol{x} - \boldsymbol{z}_i))^2 \end{aligned}$$

where $tr(\cdot)$ is the trace norm and $\boldsymbol{u}_k$ is the kth column of matrix $\mathbf{U}$. The m largest eigenvectors $\mathbf{U}_{\mathcal{P}}(\boldsymbol{x})$ of the pullback metric $\mathbf{G}^*_{\mathcal{P}}(\boldsymbol{x})$ are:

$$\begin{aligned} \mathbf{U}_{\mathcal{P}}(\boldsymbol{x}) &= \arg\max_{\mathbf{U}^T\mathbf{U}=\mathbf{I}} tr(\mathbf{U}^T \mathbf{G}^*_{\mathcal{P}}(\boldsymbol{x}) \mathbf{U}) \quad (13) \\ &= \arg\max_{\mathbf{U}^T\mathbf{U}=\mathbf{I}} \sum_{k=1}^{m} \sum_{i=1}^{n} \frac{4\bar{s}_i(\boldsymbol{x})}{\sigma^2} (\boldsymbol{u}_k^T (\boldsymbol{z}_i - E(\boldsymbol{z}_i))^2 \end{aligned}$$

where $E(\boldsymbol{z}_i) = \sum_{k=1}^{n} \bar{s}_k(\boldsymbol{x}) \boldsymbol{z}_k$

We see one key difference between $\mathbf{U}_{\mathcal{P}}(\boldsymbol{x})$ and $\mathbf{U}_{\mathcal{Q}}(\boldsymbol{x})$. In equation 13, $\mathbf{U}_{\mathcal{P}}(\boldsymbol{x})$ are the directions which maximize the sum of expected variance of $\boldsymbol{u}_k^T \boldsymbol{z}_i, k \in \{1, \ldots, m\}$, with respected to its expected mean. In contrast, the directions of $\mathbf{U}_{\mathcal{Q}}(\boldsymbol{x})$ in equation 12 maximize the sum of the unweighted "variance" of $\boldsymbol{u}_k^T s_i(\boldsymbol{x})(\boldsymbol{x} - \boldsymbol{z}_i), k \in \{1, \ldots, m\}$, without centralization. Their difference can be intuitively compared to the difference of doing local PCA with or without centralization. Therefore, $\mathbf{U}_{\mathcal{P}}(\boldsymbol{x})$ is closer to the principle directions of local anchor points. Second, since $\mathbf{G}^*_{\mathcal{P}}(\boldsymbol{x}) = \sum_{i=1}^{n} \frac{4\bar{s}_i(\boldsymbol{x})}{\sigma^2} (\boldsymbol{z}_i - E(\boldsymbol{z}_i))(\boldsymbol{z}_i - E(\boldsymbol{z}_i))^T$, it is also easy to show that $\mathbf{G}^*_{\mathcal{P}}(\boldsymbol{x})$ has no distance in the orthogonal directions of the affine subspace spanned by the weighted anchor points of $\bar{s}_i(\boldsymbol{x})\boldsymbol{z}_i$. So, $\mathbf{G}^*_{\mathcal{P}}(\boldsymbol{x})$ removes the effect of locally irrelevant dimensions to the anchor point manifold.

To show the differences of pullback metrics $\mathbf{G}^*_{\mathcal{Q}}(\boldsymbol{x})$ and $\mathbf{G}^*_{\mathcal{P}}(\boldsymbol{x})$ intuitively, we visualize their equi-distance curves in Figure 1, where the Guassian similarity function, euqation 7, is used to define the similarity maps in equations 9 and 6. As shown in Figure 1, we see that the pullback metric $\mathbf{G}^*_{\mathcal{P}}(\boldsymbol{x})$ emphasizes more the distance along the principle direction of the local anchor points than the pullback metric $\mathbf{G}^*_{\mathcal{Q}}(\boldsymbol{x})$. Furthermore, in Figure 1(b) we see that $\mathbf{G}^*_{\mathcal{P}}(\boldsymbol{x})$ has a zero distance in the direction being orthogonal to the manifold of anchor points, the straight line which the (green) anchor points lie on. Therefore, $\mathbf{G}^*_{\mathcal{P}}(\boldsymbol{x})$ is more discriminative on the manifold of the anchor points. To explore the effect of these differences, we also experimentally compare these two approaches in section 4 and the results show that learning the Fisher information distance on $\mathcal{P}$ outperforms in a significant manner learning Mahalanobis distance in proximity space $\mathcal{Q}$.

3.2. Large Margin Fisher Information Metric Learning

By applying on the learning instances the differential map f of equation (6) we map them on the statistical manifold $\mathcal{P}^{n-1}$. We are now ready to learn the Fisher information distance from the data.

Distance Parametrization. As discussed in section 2, the Fisher information distance on $\mathcal{P}^{n-1}$ can be exactly computed by the cosine distance (Lebanon, 2006; Lee et al., 2007):

$$d_{FIM}(\boldsymbol{p}^i, \boldsymbol{p}^j) = 2 \arccos(\sqrt{\boldsymbol{p}^i}^T \sqrt{\boldsymbol{p}^j}) \quad (14)$$

where $\boldsymbol{p}^i$ is the probability mass vector of the finite discrete

distribution $p^i(\xi)$. To parametrize the Fisher information distance, we apply on the probability mass vector $\boldsymbol{p}^i$ a linear transformation $\mathbf{L}$. The intuition is that, the effect of the optimal linear transformation $\mathbf{L}$ is equivalent to locating a set of hidden anchor points such that the data's similarity representation is the same as the transformed representation. Thus the parametric Fisher information distance is defined as:

$$d_{FIM}(\mathbf{L}\boldsymbol{p}^i, \mathbf{L}\boldsymbol{p}^j) = \quad 2\arccos(\sqrt{\mathbf{L}\boldsymbol{p}^i}^T \sqrt{\mathbf{L}\boldsymbol{p}^j}) \quad (15)$$
$$s.t. \quad \mathbf{L} \geq 0, \sum_i L_{ij} = 1, \forall j$$

$\mathbf{L}$ has size $k \times n$. k is the number of hidden anchor points. To speedup the learning process, in practice we often learn a low rank linear transformation matrix $\mathbf{L}$ with small k. The constraints $\mathbf{L} \geq 0$ and $\sum_i L_{ij} = 1, \forall j$ are added to ensure that each $\mathbf{L}\boldsymbol{p}^i$ is still a finite discrete distribution on the manifold $\mathcal{P}^{k-1}$.

Learning. We will follow the large margin metric learning approach of (Weinberger & Saul, 2009) and define the optimization problem of learning $\mathbf{L}$ as:

$$\min_{\mathbf{L}} \quad \sum_{ijk \in C(i,j,k)} [\epsilon_{ijk}]_+ + \alpha \sum_{i,j \to i} d_{FIM}(\mathbf{L}\boldsymbol{p}^i, \mathbf{L}\boldsymbol{p}^j) \quad (16)$$
$$s.t. \quad \mathbf{L} \geq 0$$
$$\sum_i L_{ij} = 1;\ \forall j$$
$$\epsilon_{ijk} = d_{FIM}(\mathbf{L}\boldsymbol{p}^i, \mathbf{L}\boldsymbol{p}^j) + \gamma - d_{FIM}(\mathbf{L}\boldsymbol{p}^i, \mathbf{L}\boldsymbol{p}^k)$$

where α is a parameter that balances the importance of the two terms. Unlike LMNN (Weinberger & Saul, 2009), the margin parameter γ is added in the large margin triplet constraints following the work of (Kedem et al., 2012), since the cosine distance is not linear with $\mathbf{L}^\mathbf{T}\mathbf{L}$. The large margin triplet constraints $C(i,j,k)$ for each instance $\boldsymbol{x}_i$ are generated using its k_1 same-class nearest neighbors and its k_2 different-class nearest neighbors in the $\mathcal{X}$ space and constraining the distance of each instance to its k_2 different class neighbors to be larger than those to its k_1 same class neighbors with γ margin. In the objective function of (16) the matrix $\mathbf{L}$ is learned by minimizing the sum of the hinge losses and the sum of the pairwise distances of each instance to its k_1 same-class nearest neighbors.

Optimization. Since the cosine distance defined in equation (14) is not convex, the optimization problem (16) is not convex. However, the constraints on matrix $\mathbf{L}$ are linear and we can solve this problem using a projected sub-gradient method. At each iteration, the main computation is the sub-gradient computation with complexity $O(mnk)$, where m is the number of large margin triplet constraints. n and k are the dimensions of the $\mathbf{L}$ matrix. The simplex projection operator on matrix $\mathbf{L}$ can be efficiently computed with complexity $O(nk\log(k))$ (Duchi et al., 2008). Note that, learning distance metric on $\mathcal{P}$ has been previously studied by Riemannian Metric Learning (RML) (Lebanon, 2006) and χ^2-LMNN (Kedem et al., 2012). In χ^2-LMNN, a symmetric χ^2 distance on $\mathcal{P}$ is learned with large margin idea similar to problem 16. SBFIML differs from χ^2-LMNN in that it uses the cosine distance to measure the distance on $\mathcal{P}$. As described in section 2, the cosine distance is exactly equivalent to the Fisher information distance on $\mathcal{P}$, while the χ^2 distance is only an approximation. In contrast to SBFIML and χ^2-LMNN, the work of RML focuses on unsupervised Fisher information metric learning. More importantly, both RML and χ^2-LMNN can only be applied in problems in which the input data lie on $\mathcal{P}$, while SBFIML can be applied to general data manifolds via the similarity-based differential map. Finally, note that SBFIML can also be applied to problems where we only have access to the pairwise instance similarity matrix, since it needs only the probability mass of finite discrete distributions as its input.

Local Metric Learning View of SBFIML. SBFIML can also be interpreted as a local metric learning algorithm. SBFIML defines the local metric on $\mathcal{X}$ as the pullback metric of the Fisher information metric induced by the following similarity-based parametric differential map $f_\mathbf{L} : \mathcal{X} \longrightarrow \mathcal{P}^{k-1}$:

$$f_\mathbf{L}(\boldsymbol{x}_i) = \mathbf{L} \cdot \boldsymbol{p}^i, s.t. \quad \mathbf{L} > 0, \textstyle\sum_i L_{ij} = 1, \forall j \quad (17)$$

where as before $\boldsymbol{p}^i$ is the probability mass vector of the finite discrete distribution $p^i(\xi)$ defined in equation (6). SBFIML learns the local metric by learning the parameters of $f_\mathbf{L}$. The explicit form of the pullback metric $\mathbf{G}^*$ can be computed according to the equation (5). Given the pullback metric we can approximate the geodesic distance on $\mathcal{X}$ by assuming that the geodesic curves are formed by straight lines as local metric learning methods (Noh et al., 2009; Wang et al., 2012) do, which would result in a non-metric distance. However, Lemma 2 allows us to approximate the geodesic distance on $\mathcal{X}$ by the Fisher information distance on $\mathcal{P}^{k-1}$. SBFIML follows the latter approach. Compared to the non-metric distance approximation, this new distance is a well defined distance function which has a closed form expression. Furthermore, this new distance approximation has the same asymptotic convergence result as the non-metric distance approximation.

4. Experiments

We will evaluate the performance of SBFIML on ten datasets from the UCI Machine Learning and mldata[1] repositories. The details of these datasets are reported in the first column of Table 1. All datasets are preprocessed by standardizing the input features. We compare

[1] http://mldata.org/.

Table 1. Mean and standard deviation of 5 times 10-fold CV accuracy results on $\mathbb{R}^d$ datasets. The superscripts $^{+-=}$ next to the accuracies of SBFIML indicate the result of the Student's t-test with SBMML,χ^2 LMNN, LMNN, GLML, PLML, KML and SVM. They denote respectively a significant win, loss or no difference for SBFIML. The **bold** entries for each dataset have no significant difference from the best accuracy for that dataset. The number in the parenthesis indicates the score of the respective algorithm for the given dataset based on the pairwise comparisons of the Student's t-test.

Datasets(#Inst./#Feat./#Class)	SBFIML	SBMML	χ^2 LMNN	LMNN	GLML	PLML	KML	SVM
stk25(208/172/2)	**81.6±1.8**$^{==+++==}$(5.0)	**81.7±3.0**(5.0)	**80.9±1.4**(5.0)	75.7±2.0(1.5)	72.7±1.8(0.5)	74.4±3.6(1.0)	**81.9±2.7**(5.0)	**81.2±1.0**(5.0)
wpbc(198/33/2)	**79.6±1.0**$^{==+++=+}$(5.5)	**79.3±1.1**(5.5)	**78.8±1.7**(4.5)	73.6±1.7(0.5)	76.5±1.6(3.0)	71.7±1.6(0.5)	**79.7±1.2**(5.5)	77.3±0.6(3.0)
wine(178/13/3)	**98.0±1.0**$^{===+===}$(4.0)	**98.3±0.4**(5.0)	97.4±0.3(3.5)	97.3±0.5(3.5)	96.1±1.1(0.5)	**97.5±1.1**(3.5)	**98.1±0.6**(4.0)	**98.1±0.6**(4.0)
sonar(208/60/2)	**87.2±1.3**$^{==+====}$(4.0)	**87.1±2.0**(3.5)	**86.4±2.0**(3.5)	84.8±1.5(2.0)	87.1±0.7(3.5)	86.1±1.4(3.0)	**86.9±2.2**(3.5)	**88.1±0.2**(5.0)
musk(476/166/2)	**96.1±0.4**$^{++=++++}$(6.5)	95.5±0.2(4.5)	94.8±0.5(3.0)	**95.8±0.5**(5.0)	91.3±0.6(0.5)	90.9±0.3(0.5)	95.3±0.2(4.0)	94.9±0.7(4.0)
wdbc(569/30/2)	97.2±0.4$^{-=++=-=}$(3.5)	**97.9±0.3**(6.0)	**97.5±0.5**(4.5)	96.4±0.2(1.0)	96.1±0.4(0.5)	96.8±0.5(3.0)	**97.9±0.3**(6.0)	97.3±0.2(3.5)
balance(625/4/3)	**97.5±0.5**$^{+++++++}$(6.5)	96.6±0.3(4.0)	96.2±0.5(4.0)	90.2±0.8(1.5)	88.8±0.5(0.0)	91.8±2.0(1.5)	96.6±0.3(4.0)	**97.7±0.5**(6.5)
breast(683/10/2)	**96.7±0.3**$^{==+=+==}$(4.5)	**96.4±0.5**(4.0)	**96.9±0.3**(5.0)	95.8±0.4(1.0)	96.4±0.2(3.5)	95.1±0.7(0.5)	**96.5±0.4**(4.5)	**96.9±0.2**(5.0)
australian(690/14/2)	84.6±0.3$^{+++++++-}$(6.0)	80.5±0.9(2.0)	83.5±0.5(5.0)	81.2±1.0(2.0)	80.5±0.8(2.0)	80.2±1.0(2.0)	80.8±0.6(2.0)	**85.7±0.9**(7.0)
vehicle(846/18/4)	79.2±0.6$^{+==+-+=}$(4.5)	75.7±1.1(1.0)	78.4±1.3(4.0)	79.6±0.9(4.5)	77.3±0.8(2.5)	**81.3±0.5**(6.5)	76.1±1.2(1.5)	**78.0±7.3**(3.5)
Total Score	50.0	40.5	42.0	22.5	16.5	22.0	40.0	46.5

SBFIML against three metric learning baseline methods: LMNN (Weinberger & Saul, 2009)[2], KML (Wang et al., 2011)[3], GLML (Noh et al., 2009), and PLML (Wang et al., 2012). The former two learn a global Mahalanobis metric in the input feature space $\mathbb{R}^d$ and the RKHS space respectively, and the last two learn smooth local metrics in $\mathbb{R}^d$. In addition, we also compare SBFIML against Similarity-based Mahalanobis Metric Learning (SBMML) to see the difference of pullback metrics $\mathbf{G}^*_{\mathcal{Q}}(\boldsymbol{x})$, equation 10, and $\mathbf{G}^*_{\mathcal{P}}(\boldsymbol{x})$, equation 11. SBMML learns a global Mahalanobis metric in the proximity space $\mathcal{Q}$. Similar to SBFIML, the metric is learned by optimizing the problem 16, in which the cosine distance is replaced by Mahalanobis distance. The constraints on $\mathbf{L}$ in problem 16 are also removed. To see the difference between the cosine distance used in SBFIML and the χ^2 distance used in χ^2 LMNN, we compare SBFIML against χ^2 LMNN. Note that, both methods solve exactly the same optimization problem 16 but with different distance computations. Finally, we also compare SBFIML against SVM for binary classification problems and against multi-class SVMs for multiclass classification problems. In multi-class SVMs, we use the one-against-all strategy to determine the class label.

KML, SBMML and χ^2 LMNN learn a $n \times n$ PSD matrix and are thus computationally expensive for datasets with large number of instances. To speedup the learning process, similar to SBFIML, we can learn a low rank transformation matrix $\mathbf{L}$ of size $k \times n$. For all methods, KML, SBMML, χ^2 LMNN and SBFMIL, we set $k = 0.1n$ in all experiments. The matrix $\mathbf{L}$ in KML and SBMML was initialized by clipping the $n \times n$ identity matrix into the size of $k \times n$. In a similar manner, in χ^2 LMNN and SBFIML the matrix $\mathbf{L}$ was initialized by applying on the initialization matrix $\mathbf{L}$ in KML a simplex projector which ensures the constraints in problem (16) are satisfied.

The LMNN has one hyper-parameter μ (Weinberger & Saul, 2009). We set it to its default value $\mu = 1$. As in (Noh et al., 2009), GLML uses the Gaussian distribution to model the learning instances of a given class. The hyper-parameters of PLML was set following (Wang et al., 2012). The SBFIML has two hyper-parameters α and γ. Following LMNN (Weinberger & Saul, 2009), we set the α parameter to 1. We select the margin parameter γ from $\{0.0001, 0.001, 0.01, 0.1\}$ using a 4-fold inner Cross Validation (CV). The selection of an appropriate similarity function is crucial for SBFIML. We choose the similarity function with a 4-fold inner CV from the angular similarity, equation (8), and the Gaussian similarity in equation (7). We examine two types of Gaussian similarity. In the first we set all σ_k to σ which is selected from $\{0.5\tau, \tau, 2\tau\}$, τ was set to the average of all pairwise distances. In the second we set the σ_k for each anchor point $\boldsymbol{z}_k$ separately; the σ_k was set by making the entropy of the conditional distribution $p(\boldsymbol{x}_i|\boldsymbol{z}_k) = \frac{s_k(\boldsymbol{x}_i)}{\sum_{i=1}^n s_k(\boldsymbol{x}_i)}$ equal to $\log(nc)$ (Hinton & Roweis, 2002), where n is the number of training instances and c was selected from $\{0.8, 0.9, 0.95\}$.

Since χ^2 LMNN and SBFIML apply different distance parametrizations to solve the same optimization problem, the parameters of χ^2 LMNN are set in exactly the same way as SBFIML, except that the margin parameter γ of χ^2 LMNN was selected from $\{10^{-8}, 10^{-6}, 10^{-4}, 10^{-2}\}$, because χ^2 LMNN uses the squared χ^2 distance (Kedem et al., 2012). The best similarity map for χ^2 LMNN is also selected using a 4-fold inner CV from the same similarity function set as that of SBFIML.

Akin to SBFIML, the performance of KML and SVM depends heavily on the selection of the kernel. We select automatically the best kernel with a 4-fold inner CV. The kernels are chosen from the linear, the set of polynomial (degree 2,3 and 4), the angular similarity, equation (8), and the Gaussian kernels with widths $\{0.5\tau, \tau, 2\tau\}$, as in SBFIML τ was set to the average of all pairwise distances. In addition, we also select the margin parameter γ of KML from $\{0.01, 0.1, 1, 10, 100\}$. The C parameter of SVM was se-

[2]http://www.cse.wustl.edu/∼kilian/code/code.html.

[3]http://cui.unige.ch/∼wangjun/.

Table 2. Accuracy results on large datasets.

Datasets(#Inst./#Feat./#Class)	SBFIML	SBMML	χ^2 LMNN
German(1000/20/2)	69.40$^{==}$(1.0)	69.30(1.0)	69.10(1.0)
Image(2310/18/2)	98.05$^{==}$(1.0)	98.18(1.0)	97.79(1.0)
Splice(3175/60/2)	**90.93**$^{+=}$(1.5)	90.55(0.5)	90.87(1.0)
Isolet(7797/617/26)	95.45$^{==}$(1.0)	95.19(1.0)	95.70(1.0)
Pendigits(10992/16/10)	**98.08**$^{++}$(2.0)	97.68(0.5)	97.77(0.5)
Total Score	6.5	4.0	4.5

lected from $\{0.01, 0.1, 1, 10, 100\}$. SBMML does not have any constraints on the similarity function, thus we select its similarity function with a 4-fold inner CV from a set which includes all kernel and similarity functions used in SBFIML and KML. As in KML, we select the margin parameter γ of SBMML from $\{0.01, 0.1, 1, 10, 100\}$. For all methods, except GLML and SVM which do not involve triplet constraints, the triplet constraints are constructed using three same-class and ten different-class nearest neighbors for each learning instance. Finally, we use the 1-NN rule to evaluate the performance of the different metric learning methods.

To estimate the classification accuracy we used 5 times 10-fold CV. The statistical significance of the differences were tested using Student's t-test with a p-value of 0.05. In order to get a better understanding of the relative performance of the different algorithms for a given dataset we used a simple ranking schema in which an algorithm A was assigned one point if it was found to have a statistically significantly better accuracy than another algorithm B, 0.5 points if the two algorithms did not have a significant difference, and zero points if A was found to be significantly worse than B.

Results. In Table 1 we report the accuracy results. We see that SBFIML outperforms in a statistical significant manner the single metric learning method LMNN and the local metric learning methods, GLML and PLML, in seven, eight and six out of ten datasets respectively. When we compare it to KML and SBMML, which learn a Mahalanobis metric in the RKHS and proximity space, respectively, we see that it is significantly better than KML and SBMML in four datasets and significantly worse in one dataset. Compared to χ^2 LMNN, SBFIML outperforms χ^2-LMNN on eight datasets, being statistically significant better on three, and it never loses in statistical significant manner. Finally, compared to SVM, we see that SBFIML is significantly better in two datasets and significantly worse in one dataset. In terms of the total score, SBFIML achieves the best predictive performance with 50 point, followed by SVM ,which scores 46.5 point, and χ^2-LMNN with 42 point. The local metric learning method GLML is the one that performs the worst. A potential explanation for the poor performance of GLML could be that its Gaussian distribution assumption is not that appropriate for the datasets we experimented with.

To provide a better understanding of the predictive performance difference between SBFIML, SBMML, and χ^2 LMNN, we applied them on five large datasets. To speedup the learning process, we use as anchor points 20% of randomly selected training instances. Moreover, the parameter k of low rank transformation matrix $\mathbf{L}$ was reduced to $k = 0.05n$, where n is the number of anchor points. The kernel function and similarity map was selected using 4-fold inner CV. The classification accuracy of Isolet and Pendigits are estimated by the default train and test split, for other three datasets we used 10-fold cross-validation. The statistical significance of difference were tested with McNemar's test with p-value of 0.05.

The accuracy results are reported in Table 2. We see that SBFIML achieves statistical significant better accuracy than SBMML on the two datasets, Splice and Pendigits. When compare it to χ^2 LMNN, we see it is statistical significant better on one dataset, Pendigits. In terms of total score, SBFIML achieves the best score, 6.5 points, followed by χ^2 LMNN.

5. Conclusion

In this paper we present a two-stage metric learning algorithm SBFIML. It first maps learning instances onto a statistical manifold via a similarity-based differential map and then defines the distance in the input data space by the Fisher information distance on the statistical manifold. This induces a new family of distance metrics in the input data space with two important properties. First, the induced metrics are robust to density variations in the original data space and second they have largest distance discrimination on the manifold of the anchor points. Furthermore, by learning a metric on the statistical manifold SBFIML can learn distances on different types of input feature spaces. The similarity-based map used in SBFIML is natural and flexible; unlike KML it does not need to be PSD. In addition SBFIML can be interpreted as a local metric learning method with a well defined distance approximation. The experimental results show that it outperforms in a statistical significant manner both metric learning methods and SVM.

Acknowledgments

Jun Wang was partially funded by the Swiss NSF (Grant 200021-122283/1). Ke Sun is partially supported by the Swiss State Secretariat for Education, Research and Innovation (SER grant number C11.0043). Fei Sha is supported by DARPA Award #D11AP00278 and ARO Award #W911NF-12-1-0241

Appendix

Proof of Lemma 2.

Proof. Let $\boldsymbol{\theta}$ be the coordinate of $p \in \mathcal{U} \subset \mathcal{M}^n$ under some smooth coordinate map $\varphi_{\mathcal{U}}$ and $\boldsymbol{\gamma}$ be the coordinate of $f(p) \in \mathcal{U}' \subset \mathcal{N}^m$ under some smooth coordinate map $\varphi_{\mathcal{U}'}$. Since p' approaches p, we have $\boldsymbol{\theta}' = \boldsymbol{\theta} + d\boldsymbol{\theta}$, where $\boldsymbol{\theta}'$ is the coordinate of p' under the coordinate map $\varphi_{\mathcal{U}}$ and $d\boldsymbol{\theta}$ is an infinitesimal small change approaching $\mathbf{0}$. Furthermore, since $f : \mathcal{M}^n \longrightarrow \mathcal{N}^m$ is a differential map, the function $\varphi_{\mathcal{U}'} \circ f \circ \varphi_{\mathcal{U}}^{-1} : \Theta \longrightarrow \Gamma$, which we will denote by g, is also differentiable. According to the Taylor expansion, we have $\boldsymbol{\gamma}' = g(\boldsymbol{\theta}') = g(\boldsymbol{\theta} + d\boldsymbol{\theta}) = g(\boldsymbol{\theta}) + \bigtriangledown g(\boldsymbol{\theta}) d\boldsymbol{\theta} + R_g(d\boldsymbol{\theta}, \boldsymbol{\theta}) = \boldsymbol{\gamma} + \mathbf{J} d\boldsymbol{\theta} + R_g(d\boldsymbol{\theta}, \boldsymbol{\theta})$, where $\boldsymbol{\gamma}'$ is the coordinate of $f(p')$ under the coordinate map $\varphi_{\mathcal{U}'}$, $\mathbf{J}$ is the Jacobian matrix of the function g at point $\boldsymbol{\theta}$ and $R_g(d\boldsymbol{\theta}, \boldsymbol{\theta})$ is the remainder term of linear approximation. Finally, according to the definition of pullback metric, we have $\lim_{d\boldsymbol{\theta} \to \mathbf{0}} \frac{d_{\mathbf{G}(\boldsymbol{\gamma})}(\boldsymbol{\gamma}', \boldsymbol{\gamma})}{d_{\mathbf{G}^*(\boldsymbol{\theta})}(\boldsymbol{\theta}', \boldsymbol{\theta})} = \lim_{d\boldsymbol{\theta} \to \mathbf{0}} \frac{(\mathbf{J} d\boldsymbol{\theta} + R_g(d\boldsymbol{\theta}, \boldsymbol{\theta}))^T \mathbf{G}(\boldsymbol{\gamma})(\mathbf{J} d\boldsymbol{\theta} + R_g(d\boldsymbol{\theta}, \boldsymbol{\theta}))}{d\boldsymbol{\theta}^T \mathbf{J}^T \mathbf{G}(\boldsymbol{\gamma}) \mathbf{J} d\boldsymbol{\theta}} = 1$. This ends the proof. □

References

Amari, S. and Cichocki, A. Information geometry of divergence functions. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 58(1):183–195, 2010.

Amari, S. and Nagaoka, H. *Methods of information geometry*, volume 191. Amer Mathematical Society, 2007.

Chen, Jianhui and Ye, Jieping. Training svm with indefinite kernels. In *Proceedings of the 25th international conference on Machine learning*, pp. 136–143. ACM, 2008.

Chen, Yihua, Garcia, Eric K., Gupta, Maya R., Rahimi, Ali, and Cazzanti, Luca. Similarity-based classification: Concepts and algorithms. *JMLR*, 2009.

Cuturi, M. and Avis, D. Ground metric learning. *arXiv preprint arXiv:1110.2306*, 2011.

Davis, J.V., Kulis, B., Jain, P., Sra, S., and Dhillon, I.S. Information-theoretic metric learning. In *ICML*, 2007.

Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T. Efficient projections onto the l 1-ball for learning in high dimensions. In *ICML*, 2008.

Hauberg, Sren, Freifeld, Oren, and Black, Michael. A geometric take on metric learning. In Bartlett, P., Pereira, F.C.N., Burges, C.J.C., Bottou, L., and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 25*, pp. 2033–2041, 2012.

Hinton, G. and Roweis, S. Stochastic neighbor embedding. *Advances in neural information processing systems*, 15: 833–840, 2002.

Honeine, P. and Richard, C. The angular kernel in machine learning for hyperspectral data classification. In *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2010 2nd Workshop on*, pp. 1–4. IEEE, 2010.

Jain, P., Kulis, B., and Dhillon, I. Inductive regularized learning of kernel functions. *Advances in Neural Information Processing Systems*, 23:946–954, 2010.

Kass, R.E. and Vos, P.W. *Geometrical foundations of asymptotic inference*, volume 908. Wiley-Interscience, 2011.

Kedem, Dor, Tyree, Stephen, Weinberger, Kilian, Sha, Fei, and Lanckriet, Gert. Non-linear metric learning. In Bartlett, P., Pereira, F.C.N., Burges, C.J.C., Bottou, L., and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 25*, pp. 2582–2590, 2012.

Lebanon, G. Metric learning for text documents. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4):497–508, 2006.

Lee, J.M. *Introduction to smooth manifolds*, volume 218. Springer, 2002.

Lee, S.M., Abbott, A.L., and Araman, P.A. Dimensionality reduction and clustering on statistical manifolds. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–7. IEEE, 2007.

Noh, Y.K., Zhang, B.T., and Lee, D.D. Generative local metric learning for nearest neighbor classification. *NIPS*, 2009.

Radhakrishna Rao, C. Information and accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society*, 37(3):81–91, 1945.

Van der Maaten, Laurens and Hinton, Geoffrey. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

Wang, J., Do, H., Woznica, A., and Kalousis, A. Metric learning with multiple kernels. *Advances in Neural Information Processing Systems. MIT Press*, 2011.

Wang, Jun, Kalousis, Alexandros, and Woznica, Adam. Parametric local metric learning for nearest neighbor classification. In Bartlett, P., Pereira, F.C.N., Burges, C.J.C., Bottou, L., and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 25*, pp. 1610–1618, 2012.

Weinberger, K.Q. and Saul, L.K. Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*, 10:207–244, 2009.

Ying, Y., Campbell, C., and Girolami, M. Analysis of svm with indefinite kernels. *Advances in Neural Information Processing Systems*, 22:2205–2213, 2009.