Geometry Preserving Multi-task Metric Learning

Peipei Yang, Kaizhu Huang, and Cheng-Lin Liu

National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences,
Beijing, China 100190
{ppyang,kzhuang,liucl}@nlpr.ia.ac.cn

Abstract. Multi-task learning has been widely studied in machine learning due to its capability to improve the performance of multiple related learning problems. However, few researchers have applied it on the important metric learning problem. In this paper, we propose to couple multiple related metric learning tasks with von Neumann divergence. On one hand, the novel regularized approach extends previous methods from the vector regularization to a general matrix regularization framework; on the other hand and more importantly, by exploiting von Neumann divergence as the regularizer, the new multi-task metric learning has the capability to well preserve the data geometry. This leads to more appropriate propagation of side-information among tasks and provides potential for further improving the performance. We propose the concept of *geometry preserving probability (PG)* and show that our framework leads to a larger PG in theory. In addition, our formulation proves to be jointly convex and the global optimal solution can be guaranteed. A series of experiments across very different disciplines verify that our proposed algorithm can consistently outperform the current methods.

Keywords: multi-task learning, metric learning, geometry preserving.

1 Introduction

Metric learning has been widely studied in machine learning due to its importance in many machine learning tasks [6,10]. The objective of metric learning is to learn a proper metric function from data, usually a Mahalanobis distance defined as $d_A(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^\top A(\mathbf{x}-\mathbf{y})}$, while satisfying certain extra constraints called side-information, e.g., similar (dissimilar) points should stay closer (further). On the other hand, multi-task learning (MTL), which refers to the joint training of multiple problems, has recently received considerable attention [4,8,11]. If the different problems are closely related, MTL could lead to better performance by propagating information among tasks.

Despite their importance, there are few researches combining multi-task learning with metric learning. To our best knowledge, only recently [8], [12], and [11] developed a multi-task metric learning framework separately. [8] proposed a novel multi-task framework called mtLMNN which directly extends the famous metric learning method Large Margin Nearest Neighbor (LMNN) [10]. Assuming the

P. Flach et al. (Eds.): ECML PKDD 2012, Part I, LNCS 7523, pp. 648–664, 2012.

metric of each task to be a combination of a common and a task-specific metric, mtLMNN proposed to learn the metrics jointly for all the tasks. Exploiting further the Frobenius norm as the regularization term to encourage the similarity among all tasks, mtLMNN indeed showed promising performance in several real datasets. On the other hand, [12] first concatenated all columns of each Mahalanobis matrix A_t for each task t to form a vector $\tilde{A}_t = \mathrm{vec}\,(A_t)$. Tasks are then coupled with each other by $\mathrm{tr}\left(\tilde{A}\Omega^{-1}\tilde{A}^\top\right)$ where $\tilde{A} = [\mathrm{vec}\,(A_1), \ldots, \mathrm{vec}\,(A_T)]$. The author explained this method from a probabilistic viewpoint while failing to validate it empirically. In another aspect, [11] assumed that the useful information of all tasks share a common low-rank subspace. By jointly learning the metrics in this common subspace, the performances of all tasks are improved.

All the above methods have some limitations. When describing the task relationship, the former two methods exploited merely simple vector-based divergence measures. More specifically, if we concatenated all columns of each matrix as a vector, in [8], Frobenius norm between two matrices simply presents the Euclidean distance, while, in [12], the divergence is given as the weighted Euclidean distance. Vector-based divergence may not be powerful enough to measure the relationship between matrices or distance metrics. It cannot preserve the data geometry and will lead to inaccurate information propagation among tasks. For [11], since the formulation is not convex, the global optimal solution is not guaranteed. Besides, the assumption is too strict in some cases.

For a better illustration of the above mentioned phenomenon, we show in Fig. 1 three graphs associated with different distance metrics, determined by a Mahalanobis matrix B, A_1, and A_2 respectively for each graph (from left to right). To visualize the Mahalanobis metric in the Euclidean space, we transform each point $\mathbf{x}_i$ to $\hat{\mathbf{x}}_i = A^{1/2}\mathbf{x}_i$ when plotting so that the Euclidean distance of any pair of transformed points $\|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\|^2$ is exactly the Mahalanobis distance of the original points $d_A(\mathbf{x}_i, \mathbf{x}_j)$. Geometrically observed, the metric A_2 is obviously more similar to B than to A_1. However, when calculating the similarity using the squared Frobenius norm of difference, surprisingly, A_1 is more similar to B than to A_2! This shows that minimizing Frobenius norm cannot preserve the geometry and hence it may not be appropriate for measuring the divergence of metrics.

Distinct with the above methods, in this paper, we engage the *Bregman matrix divergence* [3] and design a more general regularized framework for multi-task

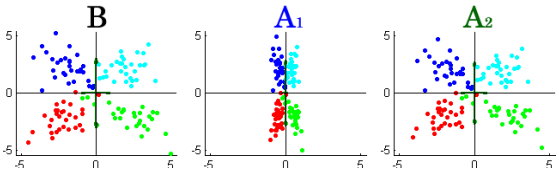


Fig. 1. Illustration of Frobenius norm for metric measurement. Using Frobenius norm, B is more similar to A_1 than to A_2, showing that Frobenius norm cannot preserve the geometry.

metric learning. On one hand, the general framework exploited a more general matrix divergence. We show that it naturally incorporates mtLMNN (using the Frobenius norm) as a special case. On the other hand and more importantly, by exploiting a special Bregman divergence called *von Neumann divergence* [3] as the regularizer, the new multi-task metric learning has the capability to well preserve the geometry when transferring information from one metric to another. We define the *geometry preserving probability* and provide theoretical analysis showing that our new multi-task metric learning method leads to a larger geometry preserving probability and has the capability to better preserve geometry. This enables more appropriate information propagation among tasks and hence provides potentials for further raising the performance. In addition to the geometry preserving property, the new multi-task framework with the von Neumann divergence remains convex, provided that any convex metric learning is used. The novel regularized multi-task metric learning framework is then justified in the probabilistic view point with a series of theoretical analysis. Extensive experimental results across very different disciplines also verify that our proposed algorithm can consistently outperform the current methods.

The rest of this paper is organized as follows. In Section 2, we will present the novel multi-task metric learning framework with Bregman matrix divergence. In Section 3, we present theoretical analysis to show our method can indeed preserve the geometry. In Section 4, we evaluate our method across five real data sets. Finally, we give concluding remarks in Section 5.

2 Novel Regularized Multi-task Metric Learning

In this section, we first present the problem definition and describe the objective of multi-task metric learning formally. Then the concept of *geometry preserving probability* is proposed to give a mathematical measure of the capability to preserve the relative distance between two metrics. After that, we introduce the main work that exploits von Neumann divergence to regularize the relationship among multiple tasks. Finally, we present a practical algorithm to solve the involved optimization problem.

2.1 Problem Definition

A *metric* defined on set $\mathbb{X}$ is a *function* $d : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_+ \doteq [0, +\infty)$ satisfying certain conditions [2]. Denoting the set containing all metrics by $\mathcal{F}_{\mathbb{X}}$ and given any pair of metrics $d_A(\cdot,\cdot), d_B(\cdot,\cdot) \in \mathcal{F}_{\mathbb{X}}$, a divergence function $D : \mathcal{F}_{\mathbb{X}} \times \mathcal{F}_{\mathbb{X}} \to \mathbb{R}_+$ is defined to measure the dissimilarity of d_A and d_B. Since the Mahalanobis metric $d_A(\cdot,\cdot)$ is ultimately determined by the Mahalanobis matrix A, we denote $D(d_A, d_B) \triangleq D(A, B)$ for short.

Assume that there are T related metric learning tasks. For each task-t, its training data set $\mathcal{S}_t$ contains N_t m-dimensional data points $\mathbf{x}_{tk} \in \mathbb{R}^m$ and a triplet set $\mathcal{T}_t = \{(i, j, k) | d(\mathbf{x}_i, \mathbf{x}_j) \leq d(\mathbf{x}_i, \mathbf{x}_k)\}$. These triplets provide side-information like relative constraints such that $\mathbf{x}_i$ is more similar to $\mathbf{x}_j$ than

to $\mathbf{x}_k$ under the new metric.[1] The objective of multi-task metric learning is to learn T proper Mahalanobis matrices A_t, $t = 1, 2, \ldots T$ jointly and simultaneously. This is significantly different from single-task metric learning where the Mahalanobis matrix is learned independently and isolatedly.

The advantages of learning multiple metrics jointly can be illustrated in Fig. 2 where the famous single task metric learning method LMNN [10] is adopted. Assume different colors indicate the labels of samples and the points with (without) a black border represent training (testing) samples. LMNN attempts to learn a new metric to encourage the neighborhood around every point to stay "pure". For each point, some points with the same label are selected as *targets* ($\triangle$) and any point with different label is expected to stand further than each target with a large margin (the dashed perimeter). Points with different label and lying within the margin are called *imposers* ($\square$). The objective of LMNN is to pull the target nearer and push all imposers outside the margin. Fig. 2(b)/2(f) show the learned metric of task-1/2 where the red/green imposers are pushed away. Unfortunately, when the training samples of green/red class are too few to represent the distribution, some testing samples invade the perimeter in the learned metric of task-1/2. However, as shown in Fig. 2(a) and 2(e), the samples in both tasks have a similar distribution to each other and we expect to improve the performance of both two tasks with help of each other. Appropriate joint metric learning of task-1 and task-2 can lead to an ideal metric for each task. For example, in Fig. 2(c)/2(g), the metric of task-1/2 can be well learned based on our novel geometry preserving framework by pushing away green/red classes with the help of task-2/1 samples. On the other hand, inappropriate multi-task metric learning may not lead to good performance. See Fig. 2(d)/2(h) for example, where the side-information is propagated by squared Frobenius norm of difference of Mahalanobis matrices as mtLMNN did.

2.2 Geometry Preserving between Metrics

In Section 2.1, we have illustrated that jointly learning multiple related metrics could benefit from the geometry preserved from other metrics. In the following, we will propose the mathematical description of the concept of *geometry preserving*.

Since the purpose of metric learning is to refine the distances among different points based on the side-information, when we mention propagating information among tasks by jointly learning multiple metric learning tasks, the information propagated is nothing but the side-information embedded in the metric. On the other hand, in most situations, the side-information specifies the relative distances between different pairs of points rather than their exact distances. For example, one popular kind of side-information is to make similar pairs nearer than dissimilar pairs. Thus, it is more important to propagate the relative distance of points from one task to another. Specifically, assume that we have two

[1] Other settings, e.g., the constraints given by similar and dissimilar pairs could be also used.

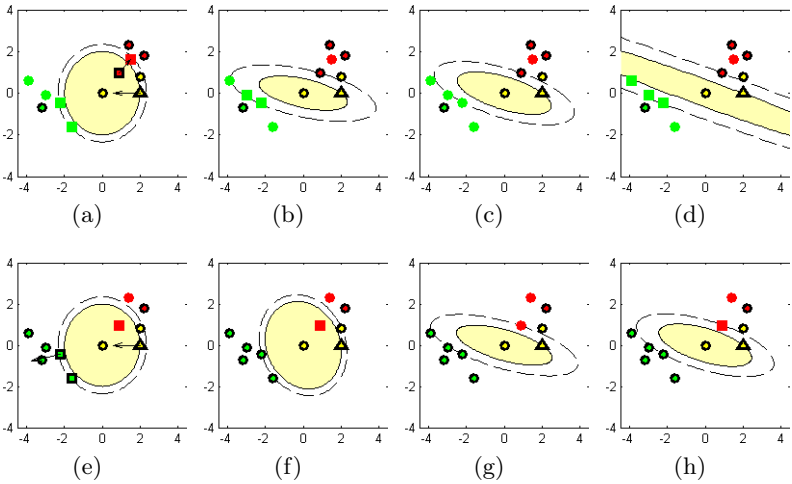


Fig. 2. An illustration of multi-task metric learning. (a/e) The original data of task 1/2. (b/f) The data of task 1/2 after single task metric learning. (c/g) The data of task 1/2 after joint metric learning using von Neumann divergence as regularizer. (d/h) The data of task 1/2 after joint metric learning using squared Frobenius norm of difference as regularizer. Joint learning of multiple tasks (given by our proposed geometry preserving framework) can lead to ideal metrics for both task-1 in (c) & task-2 in (g).

metric learning tasks to learn Mahalanobis matrices A and B respectively. Given $d_B(\mathbf{x}_1, \mathbf{x}_2) < d_B(\mathbf{x}_3, \mathbf{x}_4)$, if we are going to propagate this side-information embedded in d_B to d_A, it is desirable of d_A to make the similar judgement on the relative distance of these two pairs of points, i.e. $d_A(\mathbf{x}_1, \mathbf{x}_2) < d_A(\mathbf{x}_3, \mathbf{x}_4)$. In contrast, the exact absolute values of these distances are less important.

Based on the idea, we propose the concept of *geometry preserving probability* to measure the probability of that the relative distance of arbitrary two pairs of points can be preserved or be consistent for the two metrics.

Definition 1 (Geometry Preserving Probability). *Suppose* $\mathbf{x}_1, \mathbf{y}_1 \in \mathbb{X}$ *and* $\mathbf{x}_2, \mathbf{y}_2 \in \mathbb{X}$ *are two pairs of random points following certain distribution defined by probability density* $f(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2)$*. If two metrics* d_A *and* d_B *defined on* $\mathbb{X}$ *are used to compare the distances between each pair of points* $d(\mathbf{x}_1, \mathbf{y}_1)$ *and* $d(\mathbf{x}_2, \mathbf{y}_2)$*, the probability that* d_A *and* d_B *make the same judgement about their relative distance is called* ***geometry preserving probability*** *of* d_A *and* d_B *with* f*. It is denoted by* $PG_f(d_A, d_B)$ *with mathematical description shown in (1).*

$$\begin{aligned} PG_f(d_A, d_B) = &\mathrm{P}\left[d_A(\mathbf{x}_1, \mathbf{y}_1) > d_A(\mathbf{x}_2, \mathbf{y}_2) \ \wedge \ d_B(\mathbf{x}_1, \mathbf{y}_1) > d_B(\mathbf{x}_2, \mathbf{y}_2)\right] + \\ &\mathrm{P}\left[d_A(\mathbf{x}_1, \mathbf{y}_1) < d_A(\mathbf{x}_2, \mathbf{y}_2) \ \wedge \ d_B(\mathbf{x}_1, \mathbf{y}_1) < d_B(\mathbf{x}_2, \mathbf{y}_2)\right] \end{aligned} \tag{1}$$

where $(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2) \sim f$ *and* $\wedge$ *represents the logical* "and" *operator.*

By this definition, the larger $PG_f(d_A, d_B)$ is, the better the geometry is preserved from d_B to d_A. In the following parts, we will propose our multi-task metric

learning framework and then present the theoretical analysis, which shows that our method is more liable to make $PG_f(d_A, d_B)$ larger and thus can better preserve geometry. In contrast, mtLMNN focus more on propagating the absolute distances and could not leads to a large PG as ours.

2.3 Main Framework

We describe our novel multi-task metric learning framework as follows. Assume a common metric d_c is defined and the metric of each (the t-th) task d_t is enforced to be similar to d_c by a regularizer $D(d_t, d_c)$. Information contained in each metric can be propagated to others through the common metric. In case of the Mahalanobis metric, the regularizer can be also written as $D(A_t, B)$, where the matrices A_t and B correspond to the t-th task and the common one respectively. The novel framework can be formulated as

$$\min_{\{A_t\},B} \sum_t \left(L(A_t, \mathcal{S}_t) + \gamma D(A_t, B)\right) + \gamma_0 D(A_0, B) \quad \text{s.t. } A_t \in \mathcal{C}(\mathcal{S}_t), A_t \succeq \mathbf{0}, \quad (2)$$

where L is the loss function of the training samples of the t-th task $\mathcal{S}_t$ depending on the metric learning method, D is the divergence function to enforce the metric of the t-th task A_t similar to a common metric B, and $\mathcal{C}(\mathcal{S}_t)$ is the set of feasible A_t of the t-th task, which can be defined via side-information or the triplet set $\mathcal{T}_t$. The term $D(A_0, B)$ restricts B not far from a predefined metric A_0 as prior.

In this paper, we propose a framework to use the *Bregman matrix divergence* [3] as the regularizer $D(A, B)$ in (2), which is defined as

$$D_\phi(A, B) = \phi(A) - \phi(B) - \operatorname{tr}\left((\nabla\phi(B))^\top (A - B)\right),$$

where $\phi : \mathrm{SPD}(m) \to \mathbb{R}$ is a strictly convex, differentiable function.

It is easy to show that this framework includes mtLMNN as a special case by using $\phi(A) = \|A\|_{\mathrm{F}}^2$ and replacing $A_t \succeq \mathbf{0}$ with $A_t \succeq B \succeq \mathbf{0}$. However, this method has two main drawbacks: (1) The constraints $A_t \succeq B$ are unnecessarily strong for A_t to be a Mahalanobis matrix, which implies distance of any task has to be larger than the distance defined by the common part. (2) It is not appropriate to use Frobenius norm as the regularizer, since it cannot preserve the data geometry.

To overcome these drawbacks, we use the *von Neumann divergence* as the regularizer and obtain our multi-task metric learning method, where the von Neumann divergence is defined as $D_{\mathrm{vN}}(A, B) = \operatorname{tr}(A \log A - A \log B - A + B)$, where $\log A$ is the *matrix logarithm*[2] of A.

Using our method to learn a metric A that is assumed to be similar to B, it is more liable to obtain a solution with better geometry property preserved. We will detail the theoretical analysis in Section 3.

[2] If $A = V\Lambda V^\top$ is the eigendecomposition of A, the matrix logarithm is $V \log \Lambda V^\top$ where $\log \Lambda$ is the diagonal matrix containing the logarithm of eigenvalues.

An example. Now we revisit the example proposed in Fig. 2 where single-task metric learning fails to learn a good metric for any task. Fig. 2(c) and 2(d) show the data of task-1 in the metric learned using von Neumann divergence and Frobenius norm as regularizer respectively. Obviously, when Frobenius norm is used, although red points are pushed away, some testing points of green class invade into the margin again and the geometry has not been preserved. In contrast, when von Neumann divergence is used, both testing samples of red and green class are pushed outside the perimeter, which means the nice geometry property from task-2 is appropriately preserved after transferred to task-1. For task-2 shown in Fig. 2(g) and 2(h), von Neumann divergence also performs better than Frobenius norm.

Optimization. Since von Neumann divergence is jointly convex with two arguments [9], our multi-task metric learning method is jointly convex with its arguments if L is convex with A_t. This means that any convex metric learning method can be extended to our multi-task framework without losing its convexity. Therefore, it guarantees a global optimal solution and we solve it by alternating minimization method. Due to the convex, differentiable, and non-negative properties of von Neumann divergence, it is not difficult to verify the convergence of our algorithm.

Fix B and Optimize A_t. Suppose that L is convex with A_t, then the optimization is divided to T individual convex subproblems, each of which is a single-task metric learning problem with a regularizer. In this paper, we apply our multi-task framework to LMNN [10] metric learning approach which proved effective in many applications.

The subproblem for the t-th task can be solved by gradient descent method with $\frac{\partial \tilde{L}_t}{\partial A_t} = \frac{\partial L}{\partial A_t} + \gamma \frac{\partial D_{\text{vN}}}{\partial A_t} = \frac{\partial L}{\partial A_t} + \gamma(\log A_t - \log B)$. The first part is the gradient of a single-task metric learning problem, while the second part enforces A_t to be similar to a common matrix B.

Fix A_t and Optimize B. If all A_t are fixed, the variable to be optimized is B. With [1], the optimal solution of B is called the *Bregman representative* in case of matrix variables. It is straightforward to prove that Proposition 1 of [1] can be extended to the case of matrix and the minimizer is the weighted average of $\{A_t\}$ and A_0 as $B = (\gamma \sum_t A_t + \gamma_0 A_0)/(\gamma T + \gamma_0)$.

3 Theoretical Analysis

In this section, we analyze our multi-task metric learning method theoretically, showing how the von Neumann divergence encourages a larger geometry preserving probability and thus preserves geometry better. To this end, we firstly define an operator ρ called *scale extractor* to transform a metric to a vector called *scale vector*, which characterizes the important scale property of the metric. Since the scale vector is much more convenient to deal with than the metric which is a function, it provides a tool to bridge the von Neumann divergence and geometry

preserving probability. We establish such a relationship in three steps: (1) The geometry preserving probability monotonically decreases with a function of scale vectors $R(A, B)$. (2) For any orthonormal basis W and two Mahalanobis metrics d_A, d_B, the KL-divergence of $\rho_W(A)$ and $\rho_W(B)$ is bounded by the von Neumann divergence of A and B. (3) Minimizing $D_{\mathrm{KL}}(\rho_W(A), \rho_W(B))$ has the effect to minimize $R(A, B)$ and thus encourages larger geometry preserving probability $PG_f(A, B)$. These steps are discussed in detail in the following subsections.

3.1 Basic Definitions

Our motivation comes from the following fact. Given any pair of points $\forall \mathbf{x}, \mathbf{y} \in \mathbb{X}$, if two *metrics* d_A and d_B are similar, then the distances they give $d_A(\mathbf{x}, \mathbf{y})$ and $d_B(\mathbf{x}, \mathbf{y})$ are expected to be similar. It provides a way to measure the similarity between two metrics by comparing the distances they give for a certain pairs of points instead. Motivated by this, we can use a vector to characterize the properties of a metric and transform some problems from the intricate functional space $\mathcal{F}_{\mathbb{X}}$ to a much simpler vector space. Based on this idea, we propose the following definitions.

Definition 2 (Scale). *Given any metric $d : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_+$ and a unit vector $\mathbf{w} \in \mathbb{X}$ where $\|\mathbf{w}\| = 1$, the squared distance $d^2(\mathbf{w}, \mathbf{0})$ is defined as the **scale** of d on $\mathbf{w}$.*

Since Mahalanobis metric determines a series of scales on different directions, the essential objective of metric learning is to redefine these scales with side-information so that a certain constraints are satisfied. Due to the *translation-invariant* property of Mahalanobis metric, we always translate $\mathbf{x}$ to the original and briefly denote $d_A(\mathbf{x}, \mathbf{y}) \doteq d_A(\mathbf{z})$ where $\mathbf{z} = \mathbf{x} - \mathbf{y}$ and thus the scale of d on $\mathbf{z}$ is briefly denoted as $d^2(\mathbf{z})$.

Definition 3 (Scale Extractor). *Define the operator $\rho_W : \mathcal{F}_{\mathbb{X}} \to \mathbb{R}^n$ which transforms a metric d to a vector consisting of the scales of d on a group of vectors $W_{m\times n} = [\mathbf{w}_1\ \mathbf{w}_2\ \ldots\ \mathbf{w}_n]$ as **scale extractor**:*

$$\rho_W(d) = \left[\rho_{\mathbf{w}_1}(d)\ \rho_{\mathbf{w}_2}(d) \ldots \rho_{\mathbf{w}_n}(d)\right]^\top = \left[d^2(\mathbf{w}_1)\ d^2(\mathbf{w}_2) \ldots d^2(\mathbf{w}_n)\right]^\top$$

*The vector $\rho_W(d)$ is called the **scale vector** of d on W.*

Given any W, the more similar d_A and d_B are, the more similar $\rho_W(d_A)$ and $\rho_W(d_B)$ should be. Since $\rho_W(d_A)$ and $\rho_W(d_B)$ are just real vectors, the divergence between them is much easier to estimate and has an explicit sense as metric definition for the same points. Therefore, it can be used to define $D(d_A, d_B)$ with proper W.

When estimating the divergence of two metrics $d_A, d_B \in \mathcal{F}_{\mathbb{X}}$, a natural choice of W is an orthonormal basis of $\mathbb{X}$ because they represent the scales of d_A on different directions. Then, by enforcing $\rho_W(d_A)$ and $\rho_W(d_B)$ to be similar, we can make the scales of d_A and d_B on different directions similar. As we have indicated, in metric learning problems, we hope them to be similar in the sense

of the same relative distances. In next subsections, we will show that if we choose *KL-divergence* of $\rho_W(d_A)$ and $\rho_W(d_B)$ as the regularizer, it has the effect to encourage a larger geometry preserving probability for d_A and d_B.

3.2 Enlarging $PG_f(d_A, d_B)$ by Minimizing $R(A, B)$

In this subsection, we show that the geometry preserving probability monotonically decreases with a function of scale factor vectors of two metrics, which couples the complicated defined probability with a simpler property of metric. As we have shown in Section 2.2, the geometry preserving property is mathematically measured by the geometry preserving probability PG, whose original definition is intractable, though. In following, we propose the relationship between PG and the scale vectors which correlates $PG_f(d_A, d_B)$ with the property of d_A and d_B.

For convenience of calculating PG, we first define the *geometry preserving indicator*. In the following discussion, we always denote $\mathbf{z}_i = \mathbf{x}_i - \mathbf{y}_i$ as the difference of two points.

Definition 4 (Geometry Preserving Indicator). *The* Geometry Preserving Indicator $\Psi_{A,B}(\mathbf{x}_1 - \mathbf{y}_1, \mathbf{x}_2 - \mathbf{y}_2) = \Psi_{A,B}(\mathbf{z}_1, \mathbf{z}_2)$ *is a function that takes two metrics d_A, d_B as parameters and two differences of two pairs of points as variables. We use d_A and d_B to calculate the distances of the two pairs of points and then compare which pair is relatively further. Then $\Psi = 1$ if the two metrics give the same judgement and $\Psi = 0$ otherwise. Mathematically, it is*

$$\begin{aligned}\Psi_{A,B}(\mathbf{z}_1, \mathbf{z}_2) =& \mathbb{1}\left[(d_A(\mathbf{z}_1) > d_A(\mathbf{z}_2)) \wedge (d_B(\mathbf{z}_1) > d_B(\mathbf{z}_2))\right] + \\ & \mathbb{1}\left[(d_A(\mathbf{z}_1) < d_A(\mathbf{z}_2)) \wedge (d_B(\mathbf{z}_1) < d_B(\mathbf{z}_2))\right]\end{aligned}$$

where $\mathbb{1}[\mathcal{E}]$ is the indicator function *which equals to 1 if the logical expression $\mathcal{E}$ holds and 0 otherwise.*

Noting that any $f(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2)$ uniquely determines a probability density $\tilde{f}(\mathbf{x}_1 - \mathbf{y}_1, \mathbf{x}_2 - \mathbf{y}_2) = \tilde{f}(\mathbf{z}_1, \mathbf{z}_2)$ for the differences, the geometry preserving probability $PG_f(d_A, d_B)$ can be calculated as an integral on the whole space

$$PG_f(d_A, d_B) = \iint_{\mathbb{R}^m \times \mathbb{R}^m} \Psi_{A,B}(\mathbf{z}_1, \mathbf{z}_2) \tilde{f}(\mathbf{z}_1, \mathbf{z}_2) \mathrm{d}z_1^{(1)} \dots \mathrm{d}z_1^{(m)} \mathrm{d}z_2^{(1)} \dots \mathrm{d}z_2^{(m)} \tag{3}$$

Then we propose the theorem to couple geometry preserving probability with scales.

Theorem 1 (Geometry Preserving Theorem). *Suppose that there are two pairs of random points $\mathbf{x}_1, \mathbf{y}_1 \in \mathbb{R}^m$ and $\mathbf{x}_2, \mathbf{y}_2 \in \mathbb{R}^m$ following probability density $f(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2)$. Given any $d_B \in \mathcal{F}_{\mathbb{R}^m}$, the* geometry preserving probability *$PG_f(d_A, d_B)$ is determined by $d_A \in \mathcal{F}_{\mathbb{R}^m}$ and* ***monotonically decreases*** *with*

$$R(A, B) = \iint_{\mathbb{S}^{m-1} \times \mathbb{S}^{m-1}} R_{\mathbf{w}_1, \mathbf{w}_2}(A, B) \mathrm{d}\Omega(\mathbf{w}_1) \mathrm{d}\Omega(\mathbf{w}_2) \tag{4}$$

where

$$R_{\mathbf{w}_1,\mathbf{w}_2}(A,B)=\left|\sqrt{\frac{\rho_{\mathbf{w}_2}(A)}{\rho_{\mathbf{w}_2}(B)}}-\sqrt{\frac{\rho_{\mathbf{w}_1}(A)}{\rho_{\mathbf{w}_1}(B)}}\right|\cdot\left(\sqrt{\frac{\rho_{\mathbf{w}_1}(A)}{\rho_{\mathbf{w}_2}(B)}}+\sqrt{\frac{\rho_{\mathbf{w}_2}(A)}{\rho_{\mathbf{w}_1}(B)}}\right)^{-1}, \quad (5)$$

$\mathrm{d}\Omega(\mathbf{w}_i)$ *is the* solid angle element *corresponding to the direction of* $\mathbf{w}_i$ *which contains all the angular factors*[3] *[5], and* $\mathbb{S}^{m-1}=\{\mathbf{x}\in\mathbb{R}^m \mid \|\mathbf{x}\|=1\}$ *is the* $(m-1)$*-dimensional unit sphere in* $\mathbb{R}^m$*. The integration is calculated on* $\mathbb{S}^{m-1}$ *for both* $\mathbf{w}_1$ *and* $\mathbf{w}_2$*.*

We interpret the theorem slightly before proof. The integration (3) is taken on all the solid angle values and independent of the radius, which implies that the values of $R_{\mathbf{w}_1,\mathbf{w}_2}(A,B)$ corresponding to each pair of directions of $\mathbf{w}_1$ and $\mathbf{w}_2$ are integrated to get $R(A,B)$. Thus, if we make $R_{\mathbf{w}_1,\mathbf{w}_2}(A,B)$ smaller for each pair of directions, we can get a smaller $R(A,B)$ and a larger $PG_f(d_A,d_B)$ as we expected. What is better, this relation is independent of the distribution f. To prove the theorem, we first present a lemma.

Lemma 2. *Suppose there are two pairs of random points* $\mathbf{x}_1,\mathbf{y}_1\in\mathbb{R}^m$ *and* $\mathbf{x}_2,\mathbf{y}_2\in\mathbb{R}^m$*. For each pair, the difference* $\mathbf{z}_i=\mathbf{x}_i-\mathbf{y}_i$ *lies in a* 1-dimensional subspace $\mathbb{X}_i$ *which means there exists a unit vector* $\mathbf{w}_i\in\mathbb{X}_i$ *and a random real number* r_i *so that* $\mathbf{z}_i=r_i\mathbf{w}_i$*. Then for any Mahalanobis metrics* $d_B\in\mathcal{F}_{\mathbb{R}^m}$*, the* geometry preserving probability $PG_f(d_A,d_B)$ *is determined by* $d_A\in\mathcal{F}_{\mathbb{R}^m}$ *and* ***monotonically decreases*** *with* $R_{\mathbf{w}_1,\mathbf{w}_2}(A,B)$ *defined in (5).*

Proof. Due to the translation-invariant property of d_A and d_B, for $\forall i=1,2$, we have $d_A^2(\mathbf{x}_i,\mathbf{y}_i)=r_i^2\mathbf{w}_i^\top A\mathbf{w}_i=r_i^2\rho_{\mathbf{w}_i}(A)$, thus the squared distance d_A^2 equals to the weighted scale on $\mathbf{w}_i$ with weight r_i^2. Similarly, $d_B^2(\mathbf{x}_i,\mathbf{y}_i)=r_i^2\rho_{\mathbf{w}_i}(B)$. It is straightforward to show that

$$d_A(\mathbf{x}_1,\mathbf{y}_1)>d_A(\mathbf{x}_2,\mathbf{y}_2)\Leftrightarrow|r_1/r_2|>\sqrt{\rho_{\mathbf{w}_2}(A)/\rho_{\mathbf{w}_1}(A)} \quad (6)$$

which also holds for B. Denote $\mathbf{r}=[r_1\ r_2]^\top$ and substitute (6) into $\Psi_{A,B}$, then PG can be reformulated as a function of $\mathbf{r}$

$$PG_f(d_A,d_B)=\iint_{\mathbb{R}_+\times\mathbb{R}_+}\Psi_{A,B}(r_1\mathbf{w}_1,r_2\mathbf{w}_2)\tilde{f}(r_1,r_2)\mathrm{d}r_1\mathrm{d}r_2=\int_{S_\mathrm{I}\cup S_\mathrm{II}}\tilde{f}(\mathbf{r})\mathrm{d}\mathbf{r} \quad (7)$$

where $\tilde{f}(\mathbf{r})$ is the probability density of $\mathbf{r}$ determined by $f(\mathbf{x}_1,\mathbf{y}_1,\mathbf{x}_2,\mathbf{y}_2)$, and

$$S_\mathrm{I}=\left\{\mathbf{r}\mid|r_1/r_2|>\max\left\{\sqrt{\rho_{\mathbf{w}_2}(A)/\rho_{\mathbf{w}_1}(A)},\sqrt{\rho_{\mathbf{w}_2}(B)/\rho_{\mathbf{w}_1}(B)}\right\}\right\},$$
$$S_\mathrm{II}=\left\{\mathbf{r}\mid|r_1/r_2|<\min\left\{\sqrt{\rho_{\mathbf{w}_2}(A)/\rho_{\mathbf{w}_1}(A)},\sqrt{\rho_{\mathbf{w}_2}(B)/\rho_{\mathbf{w}_1}(B)}\right\}\right\},$$

The integral field is illustrated as the green part in Fig. 3. Since the probability density $\tilde{f}(\mathbf{r})$ is non-negative anywhere and the border corresponding to d_B is

[3] For example, for $m=2$, $\mathrm{d}\Omega(\mathbf{w}_i)=\mathrm{d}\theta$ which is independent of $\mathbf{w}_i$; for $m=3$, $\mathrm{d}\Omega(\mathbf{w}_i)=\sin\theta\mathrm{d}\theta\mathrm{d}\varphi$ where $w_i^{(1)}=\cos\theta, w_i^{(2)}=\sin\theta\cos\varphi, w_i^{(3)}=\sin\theta\sin\varphi$.

fixed, PG monotonically decreases with $|\omega|$, where ω is the angle between the two borders determined by $\rho_W(A)$ and $\rho_W(B)$. Then, we have

$$|\omega| = \left|\arctan\sqrt{\rho_{\mathbf{w}_2}(A)/\rho_{\mathbf{w}_1}(A)} - \arctan\sqrt{\rho_{\mathbf{w}_2}(B)/\rho_{\mathbf{w}_1}(B)}\right|$$
$$= \arctan\left|\frac{\sqrt{\rho_{\mathbf{w}_2}(A)/\rho_{\mathbf{w}_1}(A)} - \sqrt{\rho_{\mathbf{w}_2}(B)/\rho_{\mathbf{w}_1}(B)}}{1 + \sqrt{\rho_{\mathbf{w}_2}(A)\rho_{\mathbf{w}_2}(B)}/\sqrt{\rho_{\mathbf{w}_1}(A)\rho_{\mathbf{w}_1}(B)}}\right| = \arctan R_{\mathbf{w}_1,\mathbf{w}_2}(A,B)$$

where $R_{\mathbf{w}_1,\mathbf{w}_2}(A,B)$ is the ratio shown in (5). Since arctan is a monotony increasing function and $PG_f(d_A, d_B)$ monotonically decreases with $|\omega|$, the conclusion that $PG_f(d_A, d_B)$ monotonically decreases with $R_{\mathbf{w}_1,\mathbf{w}_2}(A,B)$ is proved. □

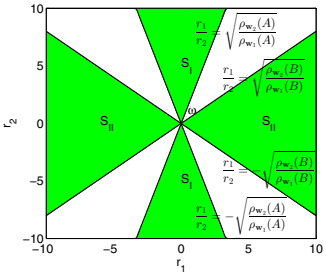
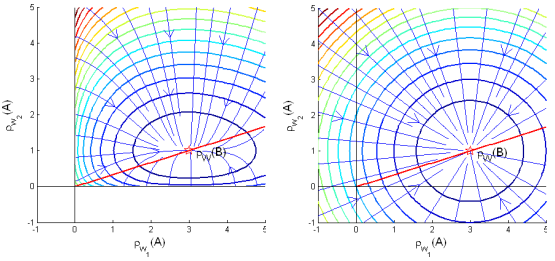


Fig. 3. Integral domain



(a) KL-divergence (b) Euclidean distance

Fig. 4. Gradient field of $D_\varphi(\rho_W(A), \rho_W(B))$.

Proof (Theorem 1). Denote $\mathbf{z}_i = r_i\mathbf{w}_i$ where $\mathbf{w}_i$ is a unit vector and r_i is the length of $\mathbf{z}_i$, then the volume element is $\mathrm{d}z_i^{(1)} \ldots \mathrm{d}z_i^{(m)} = r_i^{m-1}\mathrm{d}r_i\,\mathrm{d}\Omega(\mathbf{w}_i)$, where $\mathrm{d}\Omega(\mathbf{w}_i)$ is the solid angle corresponding to the direction $\mathbf{w}_i$.

$$\iint_{\mathbb{S}^{m-1}\times\mathbb{S}^{m-}} \iint_{\mathbb{R}_+\times\mathbb{R}_+} \Psi_{A,B}(r_1\mathbf{w}_1, r_2\mathbf{w}_2)\tilde{f}(r_1\mathbf{w}_1, r_2\mathbf{w}_2)r_1^{m-1}r_2^{m-1}\mathrm{d}r_1\mathrm{d}r_2\mathrm{d}\Omega(\mathbf{w}_1)\mathrm{d}\Omega(\mathbf{w}_2) \tag{8}$$

Note that for any fixed $\mathbf{w}_1, \mathbf{w}_2$, the inner integration is just (7) discussed in case of Lemma 2 if $\tilde{f}(r_1\mathbf{w}_1, r_2\mathbf{w}_2)r_1^{m-1}r_2^{m-1}$ is regarded as the unnormalized probability density function[4] of (r_1, r_2). Thus, replacing the inner integration of (8) with (5) and using Lemma 2, we get the conclusion that the geometry preserving probability $PG_f(d_A, d_B)$ monotonically decreases with (4). □

Remarks. Note that if d_A and d_B are learned simultaneously, PG is not guaranteed to strictly monotonically decrease with (4) because $R(A,B)$ also depends on f. However, if we have little information about f, a smaller (4) also leads to a larger PG in most cases.

[4] By the proof of Lemma 2, the conclusion also holds if f is unnormalized.

3.3 Bounding the KL-divergence with von Neumann Divergence

In this subsection, we show that by minimizing the von Neumann divergence of two Mahalanobis matrices, the KL-divergence [3] of scales on any pair of directions is minimized. This result is shown in Theorem 4 where the KL-divergence is defined as $D_{\mathrm{KL}}(\mathbf{x}, \mathbf{y}) = \sum_i x_i(\log x_i - \log y_i) - x_i + y_i$.

The main result Theorem 4 is supported by Lemma 3, a result very similar to that in quantum information [7]. We have to omit the detailed proof due to the limit of space and present only the results. We found that it can be proved in the similar way as [7].

Lemma 3. *For any trace preserving map [7] Φ, given by $\Phi(A) = \sum_{i=1}^{n} V_i A V_i^\top$ and $\sum_{i=1}^{n} V_i^\top V_i = \mathbf{I_m}$, we have that $D_{\mathrm{KL}}(\Phi(A), \Phi(B)) \leq D_{\mathrm{vN}}(A, B)$.*

Theorem 4. *Suppose $d_A, d_B \in \mathcal{F}_{\mathbb{R}^m}$ are two Mahalanobis metrics defined on $\mathbb{R}^m$, then for any orthonormal basis $W = [\mathbf{w}_1 \ \ldots \ \mathbf{w}_m]$ in $\mathbb{R}^m$, the* KL-divergence of their scale vectors $\rho_W(A)$ *and* $\rho_W(B)$ *is bounded by the* von Neumann divergence of their Mahalanobis matrices A *and* B*:* $D_{\mathrm{KL}}(\rho_W(A), \rho_W(B)) \leq D_{\mathrm{vN}}(A, B)$.

Proof. For any orthonormal basis $W = [\mathbf{w}_1 \ \ldots \ \mathbf{w}_m]$, we have

$$\begin{aligned} D_{\mathrm{KL}}\left(\rho_W(A), \rho_W(B)\right) &= \sum_i D_{\mathrm{KL}}(\mathbf{w}_i^\top A \mathbf{w}_i, \mathbf{w}_i^\top B \mathbf{w}_i) \\ &= \sum_{i,j} (\mathbf{w}_i^\top \mathbf{w}_j)^2 D_{\mathrm{KL}}(\mathbf{w}_i^\top A \mathbf{w}_i, \mathbf{w}_j^\top B \mathbf{w}_j) \\ &= D_{\mathrm{vN}}(\sum_i W_i A W_i^\top, \sum_i W_i B W_i^\top) \leq D_\phi(A, B) \end{aligned}$$

where $W_i = \mathbf{w}_i \mathbf{w}_i^\top$. The third equality is the decomposition of Bregman matrix divergence [3] and the last inequality results from Lemma 3. □

Using Theorem 4, it is easy to show that minimizing $D_{\mathrm{vN}}(A, B)$ has the effect to minimize $D_{\mathrm{KL}}(\rho_{\mathbf{w}}(A), \rho_{\mathbf{w}}(B))$ on any direction $\mathbf{w}$. Interestingly, when $D(A, B) = \|A - B\|_F^2$ is used, a similar result can be attained using simple matrix calculation. We propose it in Theorem 5 and omit the proof.

Theorem 5. *Suppose $d_A, d_B \in \mathcal{F}_{\mathbb{R}^m}$ are two Mahalanobis metrics defined on $\mathbb{R}^m$, then for any orthonormal basis $W = [\mathbf{w}_1 \ \ldots \ \mathbf{w}_m]$ in $\mathbb{R}^m$, the* squared Euclidean distance of their scales $\rho_W(A)$ *and* $\rho_W(B)$ *is bounded by the* squared Frobenius norm of the difference of their Mahalanobis matrices A *and* B*:* $\|\rho_W(A) - \rho_W(B)\|^2 \leq \|A - B\|_F^2$.

In the language of Bregman divergence, the results of Theorem 4 and Theorem 5 can be uniformly formulated as $D_\varphi(\rho_W(A), \rho_W(B)) \leq D_\phi(A, B)$, where D_φ and D_ϕ are *Bregman divergence* and *Bregman matrix divergence* with the same seed function ($\phi = \varphi \circ \lambda$). However, this result cannot be straightforwardly extended to other Bregman divergences.

3.4 Minimizing $R_{\mathbf{w}_1,\mathbf{w}_2}(A,B)$ by Minimizing $D_{\mathrm{KL}}(\rho_W(A),\rho_W(B))$

As we have proved that minimizing $D_{\mathrm{vN}}(A,B)$ is to minimize $D_{\mathrm{KL}}(\rho_W(A),\rho_W(B))$ for any W, in this subsection, we then show that it furthermore encourages a smaller $R(A,B)$ in (4) and thus a larger $PG(A,B)$ by Theorem 1.

Supposing there is a metric learning problem[5] $\min_A L(A,\mathcal{S})$ whose optimal solution is $\tilde{A}$, we have $\nabla_A L\mid_{\tilde{A}}=0$. If there exists a related task with optimal solution B and we would like to propagate the information embedded in B to A, the optimization formula becomes $\min_A L(A,\mathcal{S})+\gamma D_{\mathrm{vN}}(A,B)$ where a new loss function is added to L and the optimal solution should move to another point with a smaller loss. Obviously, it always moves towards the negative gradient direction where the loss is smaller.

Here we study how $\rho_W(A)$ is effected by the regularizer for any given W. As we have shown, when $D_{\mathrm{vN}}(A,B)$ is added to loss function, it aims to minimize $D_{\mathrm{KL}}(\rho_W(A),\rho_W(B))$ and thus $\rho_W(A)$ is more liable to move towards $-\nabla D_{\mathrm{KL}}(\rho_W(A),\rho_W(B))$. The gradient of D_{KL} with respect to $\rho_W(A)$ is

$$\nabla D_{\mathrm{KL}}=\left[\log(\rho_{\mathbf{w}_1}(A)/\rho_{\mathbf{w}_1}(B))\ \ldots\ \log(\rho_{\mathbf{w}_n}(A)/\rho_{\mathbf{w}_n}(B))\right]^\top.$$

By its formulation, the gradient on each direction $\mathbf{w}_i$ is proportional to the logarithm of the ratio of scales on $\mathbf{w}_i$. This means that the regularizer always enforces the component $\rho_{\mathbf{w}_i}(A)$ with larger $\rho_{\mathbf{w}_i}(A)/\rho_{\mathbf{w}_i}(B)$ decreases more quickly, which *encourages the ratios of scales* $\rho_{\mathbf{w}_i}(A)/\rho_{\mathbf{w}_i}(B)$ *on different* $\mathbf{w}_i$ *equal*.

Noting that the numerator of $R_{\mathbf{w}_1,\mathbf{w}_2}(A,B)$ in (5) is the absolute value of difference of the ratios of scales on two directions, encouraging $\rho_{\mathbf{w}_1}(A)/\rho_{\mathbf{w}_1}(B)=\rho_{\mathbf{w}_2}(A)/\rho_{\mathbf{w}_2}(B)$ to be equal is to minimize $R_{\mathbf{w}_1,\mathbf{w}_2}(A,B)$. Thus the main conclusion of this subsection can be proposed as

Proposition 1. *For any n unit vectors $W=[\mathbf{w}_1\ \ldots\ \mathbf{w}_n]\in\mathbb{R}^{m\times n}$, the regularizer $\min_{\rho_W(A)} D_{\mathrm{KL}}(\rho_W(A),\rho_W(B))$ encourages the solution to make $R_{\mathbf{w}_i,\mathbf{w}_j}(A,B)$ smaller for $\forall i,j$.*

In contrast, if $D(A,B)=\|A-B\|_F^2$ is used, the equivalent regularizer is $\|\rho_W(A)-\rho_W(B)\|^2$. It *encourages the differences of scales* $(\rho_{\mathbf{w}_i}(A_t)-\rho_{\mathbf{w}_i}(B))$ *on different* $\mathbf{w}_i$ *equal*, which is not beneficial to minimizing $R_{\mathbf{w}_i,\mathbf{w}_j}(A,B)$.

This phenomenon is illustrated with the contour and gradient field in Fig. 4 where the red line represents the points with the same ratio of scales. The concentric circles are contours of $D_\varphi(\rho_W(A),\rho_W(B))$ and the radial lines are field lines of its negative gradient where the tangent direction at any point of the line indicates $-\nabla_{\rho_W(A)}D_\varphi(\rho_W(A),\rho_W(B))$. Minimizing $D_\varphi(\rho_W(A),\rho_W(B))$ with respect to $\rho_W(A)$ will make the solution move along the gradient field lines because it directs to the steepest descendent direction. From this figure, we see that the field lines in Fig. 4(a) are more liable to go towards the red line, which makes the solution of $\rho_{\mathbf{w}_1}(A)/\rho_{\mathbf{w}_1}(B)$ more similar to $\rho_{\mathbf{w}_2}(A)/\rho_{\mathbf{w}_2}(B)$.

[5] The constraints can be reformulated into loss function using Lagrangian multiplier.

3.5 Summary

As a short summary of previous theoretical analysis, we have Proposition 2 for our multi-task metric learning framework.

Proposition 2 (Geometry Preserving with von Neumann divergence). *When the von Neumann divergence is minimized, the KL-divergence of the scales on different directions is minimized. This makes a smaller* $R_{\mathbf{w}_1,\mathbf{w}_2}(A,B)$ *for any pair of directions and thus a smaller* $R(A,B)$*, further leading to a larger* $PG_f(d_A,d_B)$ *by Theorem 1. In short, von Neumann divergence* $D_{\mathrm{vN}}(A,B)$ *encourages a larger* $PG_f(A,B)$ *and can thus better propagate the side-information about relative distance.*

4 Experiments

In this section, we conduct a series of experiments to validate the advantages of our proposed approach. In our experiments, we choose LMNN [10] as the metric learning algorithm for all methods which determines the loss function L in (2). For brevity, we call our proposed multi-task metric learning with von Neumann divergence as *mt-von*, while the method proposed in [8] is written in short as *mt-Frob* (also called mtLMNN). We compare them with three baseline methods: the *Euclidean* metric, the single-task metric learning (in short *stLMNN*) and the uniform task metric learning (in short *utLMNN*). stLMNN means that a metric is learned for each task independently, while utLMNN puts the samples of all tasks together and train a uniform metric for all tasks.

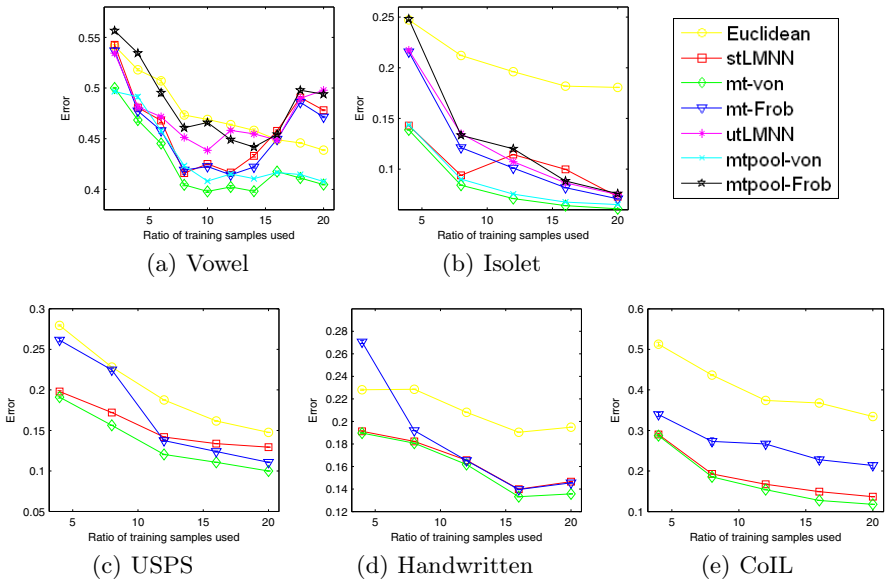
We learn a specific metric using different methods. According to the distances calculated based on the learned metric, we use 1-Nearest Neighbor as the final classifier to predict the label of a new test sample. If all tasks share a common label space, which is referred as the *label-compatible* scenario [8], we also evaluate with the *pooled training sets* [8] at the classification phase. This special classification setting is called *mtpool-von* or *mtpool-Frob*, depending on the regularizer. We also report the performance of nearest neighbor using the Euclidean distance (in short *Euclidean*) as the baseline. We tune the hyper-parameters involved in LMNN by cross validation.

We evaluate the above mentioned methods on five real data sets obtained from very different disciplines. (1). **Handwritten Letter Classification** dataset[6] consists of 8 binary handwritten letter classification problems. Each classification problem is regarded as one task. Some randomly selected samples are used to train a metric while the remaining for test. (2). **USPS digit** dataset[7] consists of 7,291 16×16 grayscale images of digits $0 \sim 9$. For each digit, we can get a two-class classification task in which the samples of this digit represent the positive patterns and the others negative patterns. (3). **Isolet** dataset[8] was collected from 150 speakers uttering all characters in the English alphabet twice. The task is

[6] `http://multitask.cs.berkeley.edu/`

[7] `http://www-i6.informatik.rwth-aachen.de/~keysers/usps.html`

[8] Available from UCI Machine Learning Repository.



(a) Vowel (b) Isolet

(c) USPS (d) Handwritten (e) CoIL

Fig. 5. Experiment results on five datasets

to classify the letter to be uttered. The speakers are grouped into 5 smaller sets of similar speakers and this makes the problem naturally be suitable for multi-task learning. Each subset is treated as a task and they are trained jointly. (4). **Insurance Company** (CoIL) Benchmark dataset[9] contains information on customers of an insurance company. The data consist of 86 variables. We select out the $68 \sim 73$-th variables as categorical features and predict their values with other features. (5). **Multi-speaker Vowel Classification** dataset[10] consists of 11 vowels uttered by 15 speakers of British English. We used the data of 1-8 (9-15) speakers as the training (testing) set. In both of them, speakers are divided into two subgroups according to their gender. It is reasonable because men pronounce in a different style with women. For this dataset, we treat each subgroup as a task.

For the first 4 datasets, we randomly choose a certain number of samples as the training set and leave the remaining samples as the test set. For the Multi-speaker Vowel dataset, we randomly select a number of samples from the 1-8 speakers as the training samples, and consider all the samples from the 9-15 speakers as the test set. In each experiment, we vary the number of training samples in each class from 4 to 20 and repeat the evaluations 10 times. The average error rates over all the tasks and the 10 times evaluations are reported in Fig. 5 as the final results. Note that, similar to [8], the five datasets are categorized into *label-compatible* and *label-incompatible* according to whether all

[9] `http://kdd.ics.uci.edu/databases/tic/tic.html`

[10] Available from UCI Machine Learning Repository.

tasks share a common label space. For label-compatible datasets, we compare all approaches mentioned above; for label-incompatible datasets, since tasks have different label spaces and $\bigcup \mathcal{S}_\tau$ is meaningless, the utLMNN, mtpool-von, and mtpool-Frob are not evaluated.

Observed from the experimental results, our proposed multi-task metric learning method performs the best across all the data sets whatever the number of training samples are used. This clearly demonstrates the superiority of our proposed multi-task framework. In particular, the geometry preserving mt-von method demonstrated significantly better performance against mt-Frob or mtLMNN consistently in all the cases. This clearly validates that the performance can be improved by preserving relative distances. For the label-compatible datasets, we see that in most cases, the performance is better if only the training samples in the task are used as the prototype of k-NN classifier. This once again demonstrates the advantages of our proposed method.

5 Conclusion

In this paper, we propose a novel multi-task metric learning framework using von Neumann divergence. On one hand, the novel regularized approach extends previous methods from the vector regularization to a general matrix regularization framework; on the other hand and more importantly, by exploiting von Neumann divergence as the regularizer, the new multi-task metric learning has the capability to well preserve the data geometry. This leads to more appropriate propagation of side-information among tasks and proves very important for further improving the performance. We propose the concept of *geometry preserving probability (PG)* and justify our framework with a series of theoretical analysis. Furthermore, our formulation is jointly convex and the global optimal solution can be guaranteed. A series of experiments verify that our proposed algorithm can significantly outperform the current methods.

Acknowledgements. This work has been supported in part by the National Basic Research Program of China (973 Program) Grant 2012CB316301, the National Natural Science Foundation of China (NSFC) Grants 61075052 and 60825301, and Tsinghua National Laboratory for Information Science and Technology (TNList) Cross-discipline Foundation.

References

1. Banerjee, A., Merugu, S., Dhillon, I.S., Ghosh, J.: Clustering with bregman divergences. Journal of Machine Learning Research 6, 1705–1749 (2005)
2. Burago, D., Burago, Y., Ivanov, S.: A Course in Metric Geometry. American Mathematical Society (June 2001)
3. Dhillon, I.S., Tropp, J.A.: Matrix nearness problems with bregman divergences. SIAM Journal on Matrix Analysis and Applications 29, 1120–1146 (2008)

4. Evgeniou, T., Micchelli, C.A., Pontil, M.: Learning multiple tasks with kernel methods. Journal of Machine Learning Research 6, 615–637 (2005)
5. Haber, H.E.: The volume and surface area of ann-dimensional hypersphere (2011), http://scipp.ucsc.edu/~haber/ph116A/volume_11.pdf
6. Huang, K., Ying, Y., Campbell, C.: Generalized sparse metric learning with relative comparisons. Knowledge and Information Systems 28(1), 25–45 (2011)
7. Lindblad, G.: Completely positive maps and entropy inequalities. Commun. Math. Phys. 40(2), 147–151 (1975)
8. Parameswaran, S., Weinberger, K.: Large margin multi-task metric learning. In: Advances in Neural Information Processing Systems 23, pp. 1867–1875 (2010)
9. Tropp, J.A.: From joint convexity of quantum relative entropy to a concavity theorem of lieb. Proceedings of the American Mathematical Society 140, 1757–1760 (2012)
10. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research 10, 207–244 (2009)
11. Yang, P., Huang, K., Liu, C.L.: A multi-task framework for metric learning with common subspace. Neural Computing and Applications, 1–11 (2012)
12. Zhang, Y., Yeung, D.Y.: Transfer metric learning by learning task relationships. In: Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2010)