Sparse Compositional Metric Learning

Yuan Shi[*] and **Aurélien Bellet**[*] and **Fei Sha**
Department of Computer Science
University of Southern California
Los Angeles, CA 90089, USA
{yuanshi,bellet,feisha}@usc.edu

Abstract

We propose a new approach for metric learning by framing it as learning a sparse combination of locally discriminative metrics that are inexpensive to generate from the training data. This flexible framework allows us to naturally derive formulations for global, multi-task and local metric learning. The resulting algorithms have several advantages over existing methods in the literature: a much smaller number of parameters to be estimated and a principled way to generalize learned metrics to new testing data points. To analyze the approach theoretically, we derive a generalization bound that justifies the sparse combination. Empirically, we evaluate our algorithms on several datasets against state-of-the-art metric learning methods. The results are consistent with our theoretical findings and demonstrate the superiority of our approach in terms of classification performance and scalability.

Introduction

The need for measuring distance or similarity between data instances is ubiquitous in machine learning and many application domains. However, each problem has its own underlying semantic space for defining distances that standard metrics (e.g., the Euclidean distance) often fail to capture. This has led to a growing interest in *metric learning* for the past few years, as summarized in two recent surveys (Bellet, Habrard, and Sebban 2013; Kulis 2012). Among these methods, learning a globally linear Mahalanobis distance is by far the most studied setting. Representative methods include (Xing et al. 2002; Goldberger et al. 2004; Davis et al. 2007; Jain et al. 2008; Weinberger and Saul 2009; Shen et al. 2012; Ying and Li 2012). This is equivalent to learning a linear projection of the data to a feature space where constraints on the training set (such as "$\boldsymbol{x}_i$ should be closer to $\boldsymbol{x}_j$ than to $\boldsymbol{x}_k$") are better satisfied.

Although the performance of these learned metrics is typically superior to that of standard metrics in practice, a single linear metric is often unable to accurately capture the complexity of the task, for instance when the data are multimodal or the decision boundary is complex. To overcome this limitation, recent work has focused on learning *multiple locally linear metrics* at several locations of the feature space (Frome et al. 2007; Weinberger and Saul 2009; Zhan et al. 2009; Hong et al. 2011; Wang, Woznica, and Kalousis 2012), to the extreme of learning one metric per training instance (Noh, Zhang, and Lee 2010). This line of research is motivated by the fact that locally, simple linear metrics perform well (Ramanan and Baker 2011; Hauberg, Freifeld, and Black 2012). The main challenge is to integrate these metrics into a meaningful global one while keeping the number of learning parameters to a reasonable level in order to avoid heavy computational burden and severe overfitting. So far, existing methods are not able to compute valid (smooth) global metrics from the local metrics they learn and do not provide a principled way of generalizing to new regions of the space at test time. Furthermore, they scale poorly with the dimensionality D of the data: typically, learning a Mahalanobis distance requires $O(D^2)$ parameters and the optimization involves projections onto the positive semidefinite cone that scale in $O(D^3)$. This is expensive even for a single metric when D is moderately large.

In this paper, we study metric learning from a new perspective to efficiently address these key challenges. We propose to learn metrics as *sparse compositions of locally discriminative metrics*. These "basis metrics" are low-rank and extracted efficiently from the training data at different local regions, for instance using Fisher discriminant analysis. Learning higher-rank linear metrics is then formulated as learning the combining weights, using sparsity-inducing regularizers to select only the most useful basis elements. This provides a unified framework for metric learning, as illustrated in Figure 1, that we call SCML (for Sparse Compositional Metric Learning). In SCML, the number of parameters to learn is much smaller than existing approaches and projections onto the positive semidefinite cone are not needed. This gives an efficient and flexible way to learn a single global metric when D is large.

The proposed framework also applies to multi-task metric learning, where one wants to learn a global metric for several related tasks while exploiting commonalities between them (Caruana 1997; Parameswaran and Weinberger 2010). This is done in a natural way by means of a group sparsity regularizer that makes the task-specific metrics share the same basis subset. Our last and arguably most interesting

[*] Equal contribution.

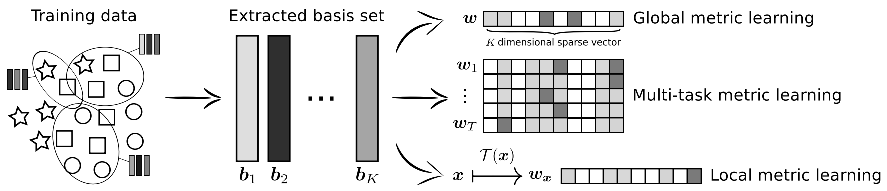


Figure 1: Illustration of the general framework and its applications. We extract locally discriminative basis elements from the training data and cast metric learning as learning sparse combinations of these elements. We formulate global metric learning as learning a single sparse weight vector $\boldsymbol{w}$. For multi-task metric learning, we learn a vector $\boldsymbol{w}_t$ for each task where all tasks share the same basis subset. For local metric learning we learn a function $\mathcal{T}(\boldsymbol{x})$ that maps any instance $\boldsymbol{x}$ to its associated sparse weight vector $\boldsymbol{w}_{\boldsymbol{x}}$. Shades of grey encode weight magnitudes.

contribution is a new formulation for local metric learning, where we learn a transformation $\mathcal{T}(\boldsymbol{x})$ that takes as input any instance $\boldsymbol{x}$ and outputs a sparse weight vector defining its metric. This can be seen as learning a smoothly varying metric tensor over the feature space (Ramanan and Baker 2011; Hauberg, Freifeld, and Black 2012). To the best of our knowledge, it is the first discriminative metric learning approach capable of computing an instance-specific metric for *any* point in a principled way. All formulations can be solved using scalable optimization procedures based on stochastic subgradient descent with proximal operators (Duchi and Singer 2009; Xiao 2010).

We present both theoretical and experimental evidence supporting the proposed approach. We derive a generalization bound which provides a theoretical justification to seeking sparse combinations and suggests that the basis set B can be large without incurring overfitting. Empirically, we evaluate our algorithms against state-of-the-art global, local and multi-task metric learning methods on several datasets. The results strongly support the proposed framework.

Proposed Approach

In this section, we present the main idea of sparse compositional metric learning (SCML) and show how it can be used to unify several existing metric learning paradigms and lead to efficient new formulations.

Main Idea

We assume the data lie in $\mathbb{R}^D$ and focus on learning (squared) Mahalanobis distances $d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x} - \boldsymbol{x}')^{\mathrm{T}} \boldsymbol{M} (\boldsymbol{x} - \boldsymbol{x}')$ parameterized by a positive semidefinite (PSD) $D \times D$ matrix $\boldsymbol{M}$. Note that $\boldsymbol{M}$ can be represented as a nonnegative weighted sum of K rank-1 PSD matrices:[1]

$$\boldsymbol{M} = \sum_{i=1}^{K} w_i \boldsymbol{b}_i \boldsymbol{b}_i^{\mathrm{T}}, \quad \text{with } \boldsymbol{w} \geq 0, \qquad (1)$$

where the $\boldsymbol{b}_i$'s are D-dimensional column vectors.

In this paper, we use the form (1) to cast metric learning as learning a *sparse combination of basis elements* taken from a basis set $B = \{\boldsymbol{b}_i\}_{i=1}^{K}$. The key to our framework is the fact that such a B is made readily available to the algorithm and consists of rank-one metrics that are *locally discriminative*. Such basis elements can be easily generated from the training data at several local regions — in the experiments, we simply use Fisher discriminant analysis (see the corresponding section for details). They can then be combined to form a single global metric, multiple global metrics (in the multi-task setting) or a metric tensor (implicitly defining an infinite number of local metrics) that varies smoothly across the feature space, as we will show in later sections.

We use the notation $d_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{x}')$ to highlight our parameterization of the Mahalanobis distance by $\boldsymbol{w}$. Learning $\boldsymbol{M}$ in this form makes it PSD by design (as a nonnegative sum of PSD matrices) and involves K parameters (instead of D^2 in most metric learning methods), enabling it to more easily deal with high-dimensional problems. We also want the combination to be *sparse*, i.e., some w_i's are zero and thus $\boldsymbol{M}$ only depends on a small subset of B. This provides some form of regularization (as shown later in Theorem 1) as well as a way to tie metrics together when learning multiple metrics. In the rest of this section, we apply the proposed framework to several metric learning paradigms (see Figure 1).

Global Metric Learning

In global metric learning, one seeks to learn a single metric $d_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{x}')$ from a set of distance constraints on the training data. Here, we use a set of triplet constraints C where each $(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k) \in C$ indicates that the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ should be smaller than the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_k$. C may be constructed from label information, as in LMNN (Weinberger and Saul 2009), or in an unsupervised manner based for instance on implicit users' feedback (such as clicks on search engine results). Our formulation for global metric learning, SCML-Global, is simply to combine the local basis elements into a higher-rank global metric that satisfies well the constraints in C:

$$\min_{\boldsymbol{w}} \frac{1}{|C|} \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k) \in C} L_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k) + \beta \|\boldsymbol{w}\|_1, \qquad (2)$$

where $L_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k) = [1 + d_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{x}_j) - d_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{x}_k)]_+$ with $[\cdot]_+ = \max(0, \cdot)$, and $\beta \geq 0$ is a regularization parameter. The first term in (2) is the classic margin-based hinge loss function. The second term $\|\boldsymbol{w}\|_1 = \sum_{i=1}^{K} w_i$ is the ℓ_1 norm regularization which encourages sparse solutions, allowing the selection of relevant basis elements. SCML-Global is convex by the linearity of both terms and is bounded below, thus it has a global minimum.

[1] Such an expression exists for any PSD matrix $\boldsymbol{M}$ since the eigenvalue decomposition of $\boldsymbol{M}$ is of the form (1).

Multi-Task Metric Learning

Multi-task learning (Caruana 1997) is a paradigm for learning several tasks simultaneously, exploiting their commonalities. When tasks are related, this can perform better than separately learning each task. Recently, multi-task learning methods have successfully built on the assumption that the tasks should share a common low-dimensional representation (Argyriou, Evgeniou, and Pontil 2008; Yang, Kim, and Xing 2009; Gong, Ye, and Zhang 2012). In general, it is unclear how to achieve this in metric learning. In contrast, learning metrics as sparse combinations allows a direct translation of this idea to multi-task metric learning.

Formally, we are given T different but somehow related tasks with associated constraint sets $C_1, \dots, C_T$ and we aim at learning a metric $d_{\boldsymbol{w}_t}(\boldsymbol{x}, \boldsymbol{x}')$ for each task t while sharing information across tasks. In the following, the basis set B is the union of the basis sets $B_1, \dots, B_T$ extracted from each task t. Our formulation for multi-task metric learning, mt-SCML, is as follows:

$$\min_{\boldsymbol{W}} \sum_{t=1}^{T} \frac{1}{|C_t|} \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j,\boldsymbol{x}_k)\in C_t} L_{\boldsymbol{w}_t}(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k) + \beta \|\boldsymbol{W}\|_{2,1},$$

where $\boldsymbol{W}$ is a $T \times K$ nonnegative matrix whose t-th row is the weight vector $\boldsymbol{w}_t$ defining the metric for task t, $L_{\boldsymbol{w}_t}(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k) = [1 + d_{\boldsymbol{w}_t}(\boldsymbol{x}_i, \boldsymbol{x}_j) - d_{\boldsymbol{w}_t}(\boldsymbol{x}_i, \boldsymbol{x}_k)]_+$ and $\|\boldsymbol{W}\|_{2,1}$ is the ℓ_2/ℓ_1 mixed norm used in the group lasso problem (Yuan and Lin 2006). It corresponds to the ℓ_1 norm applied to the ℓ_2 norm of the columns of $\boldsymbol{W}$ and is known to induce group sparsity at the column level. In other words, this regularization makes most basis elements either have zero weight or nonzero weight *for all tasks*.

Overall, while each metric remains task-specific ($d_{\boldsymbol{w}_t}$ is only required to satisfy well the constraints in C_t), it is composed of *shared features* (i.e., it potentially benefits from basis elements generated from other tasks) that are regularized to be relevant *across tasks* (as favored by the group sparsity). As a result, all learned metrics can be expressed as combinations of the same basis subset of B, though with different weights for each task. Since the ℓ_2/ℓ_1 norm is convex, mt-SCML is again convex.

Local Metric Learning

Local metric learning addresses the limitations of global methods in capturing complex data patterns (Frome et al. 2007; Weinberger and Saul 2009; Zhan et al. 2009; Noh, Zhang, and Lee 2010; Hong et al. 2011; Wang, Woznica, and Kalousis 2012). For heterogeneous data, allowing the metric to vary across the feature space can capture the semantic distance much better. On the other hand, local metric learning is costly and often suffers from severe overfitting since the number of parameters to learn can be very large. In the following, we show how our framework can be used to derive an efficient local metric learning method.

We aim at learning a *metric tensor* $\mathcal{T}(\boldsymbol{x})$, which is a smooth function that (informally) maps any instance $\boldsymbol{x}$ to its metric matrix (Ramanan and Baker 2011; Hauberg, Freifeld, and Black 2012). The distance between two points should then be defined as the geodesic distance on a Riemannian manifold. However, this requires solving an intractable problem, so we use the widely-adopted simplification that distances from point $\boldsymbol{x}$ are computed based on its own metric alone (Zhan et al. 2009; Noh, Zhang, and Lee 2010; Wang, Woznica, and Kalousis 2012):

$$\begin{aligned} d_{\mathcal{T}}(\boldsymbol{x}, \boldsymbol{x}') &= (\boldsymbol{x} - \boldsymbol{x}')^{\mathrm{T}} \mathcal{T}(\boldsymbol{x}) (\boldsymbol{x} - \boldsymbol{x}') \\ &= (\boldsymbol{x} - \boldsymbol{x}')^{\mathrm{T}} \sum_{i=1}^{K} w_{\boldsymbol{x},i} \boldsymbol{b}_i \boldsymbol{b}_i^{\mathrm{T}} (\boldsymbol{x} - \boldsymbol{x}'), \end{aligned}$$

where $\boldsymbol{w}_{\boldsymbol{x}}$ is the weight vector for instance $\boldsymbol{x}$.

We could learn a weight vector for each training point. This would result in a formulation similar to mt-SCML, where each training instance is considered as a task. However, in the context of local metric learning, this is not an appealing solution. Indeed, for a training sample of size S we would need to learn SK parameters, which is computationally difficult and leads to heavy overfitting for large-scale problems. Furthermore, this gives no principled way of computing the weight vector of a test instance.

We instead propose a more effective solution by constraining the weight vector for an instance $\boldsymbol{x}$ to parametrically depend on some embedding of $\boldsymbol{x}$:

$$\mathcal{T}_{\boldsymbol{A},\boldsymbol{c}}(\boldsymbol{x}) = \sum_{i=1}^{K} (\boldsymbol{a}_{\boldsymbol{i}}^{\mathrm{T}} \boldsymbol{z}_{\boldsymbol{x}} + c_i)^2 \boldsymbol{b}_i \boldsymbol{b}_i^{\mathrm{T}}, \tag{3}$$

where $\boldsymbol{z}_{\boldsymbol{x}} \in D'$ is an embedding of $\boldsymbol{x}$,[2] $\boldsymbol{A} = [\boldsymbol{a_1} \dots \boldsymbol{a_K}]^{\mathrm{T}}$ is a $D' \times K$ real-valued matrix and $\boldsymbol{c} \in \mathbb{R}^K$. The square makes the weights nonnegative $\forall \boldsymbol{x} \in \mathbb{R}^D$, ensuring that they define a valid (pseudo) metric. Intuitively, (3) combines the locally discriminative metrics with weights that depend on the position of the instance in the feature space.

There are several advantages to this formulation. First, by learning $\boldsymbol{A}$ and $\boldsymbol{c}$ we implicitly learn a different metric not only for the training data but for any point in the feature space. Second, if the embedding is smooth, $\mathcal{T}_{\boldsymbol{A},\boldsymbol{c}}(\boldsymbol{x})$ is a smooth function of $\boldsymbol{x}$, therefore similar instances are assigned similar weights. This can be seen as some kind of manifold regularization. Third, the number of parameters to learn is now $K(D'+1)$, thus independent of both the size of the training sample and the dimensionality of $\boldsymbol{x}$. Our formulation for local metric learning, SCML-Local, is as follows:

$$\min_{\tilde{\boldsymbol{A}}} \frac{1}{|C|} \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j,\boldsymbol{x}_k)\in C} L_{\mathcal{T}_{\boldsymbol{A},\boldsymbol{c}}}(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k) + \beta \|\tilde{\boldsymbol{A}}\|_{2,1},$$

where $\tilde{\boldsymbol{A}}$ is a $(D'+1) \times K$ matrix denoting the concatenation of $\boldsymbol{A}$ and $\boldsymbol{c}$, and $L_{\mathcal{T}_{\boldsymbol{A},\boldsymbol{c}}}(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k) = \left[1 + d_{\mathcal{T}_{\boldsymbol{A},\boldsymbol{c}}}(\boldsymbol{x}_i, \boldsymbol{x}_j) - d_{\mathcal{T}_{\boldsymbol{A},\boldsymbol{c}}}(\boldsymbol{x}_i, \boldsymbol{x}_k)\right]_+$. The ℓ_2/ℓ_1 norm on $\tilde{\boldsymbol{A}}$ introduces sparsity at the column level, regularizing the local metrics to use the same basis subset. Interestingly, if $\boldsymbol{A}$ is the zero matrix, we recover SCML-Global. SCML-Local is nonconvex and is thus subject to local minima.

[2] In our experiments, we use kernel PCA (Schölkopf, Smola, and Müller 1998) as it provides a simple way to limit the dimension and thus the number of parameters to learn. We use RBF kernel with bandwidth set to the median Euclidean distance in the data.

Optimization

Our formulations use (nonsmooth) sparsity-inducing regularizers and typically involve a large number of triplet constraints. We can solve them efficiently using stochastic composite optimization (Duchi and Singer 2009; Xiao 2010), which alternates between a stochastic subgradient step on the hinge loss term and a proximal operator (for ℓ_1 or $\ell_{2,1}$ norm) that explicitly induces sparsity. We solve SCML-Global and mt-SCML using Regularized Dual Averaging (Xiao 2010), which offers fast convergence and levels of sparsity in the solution comparable to batch algorithms. For SCML-Local, due to local minima, we ensure improvement over the optimal solution $\boldsymbol{w}^*$ of SCML-Global by using a forward-backward algorithm (Duchi and Singer 2009) which is initialized with $\boldsymbol{A} = \mathbf{0}$ and $c_i = \sqrt{w_i^*}$.

Recall that unlike most existing metric learning algorithms, we do not need to perform projections onto the PSD cone, which scale in $O(D^3)$ for a $D \times D$ matrix. Our algorithms thereby have a significant computational advantage for high-dimensional problems.

Theoretical Analysis

In this section, we provide a theoretical analysis of our approach in the form of a generalization bound based on algorithmic robustness analysis (Xu and Mannor 2012) and its adaptation to metric learning (Bellet and Habrard 2012). For simplicity, we focus on SCML-Global, our global metric learning formulation described in (2).

Consider the supervised learning setting, where we are given a labeled training sample $S = \{\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i)\}_{i=1}^n$ drawn i.i.d. from some unknown distribution P over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. We call a triplet $(\boldsymbol{z}, \boldsymbol{z}', \boldsymbol{z}'')$ *admissible* if $y = y' \neq y''$. Let C be the set of admissible triplets built from S and $L(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{z}', \boldsymbol{z}'') = [1 + d_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{x}') - d_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{x}'')]_+$ denote the loss function used in (2), with the convention that L returns 0 for non-admissible triplets.

Let us define the *empirical loss* of $\boldsymbol{w}$ on S as

$$\mathcal{R}^S_{emp}(\boldsymbol{w}) = \frac{1}{|C|} \sum_{(\boldsymbol{z},\boldsymbol{z}',\boldsymbol{z}'') \in C} L(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{z}', \boldsymbol{z}''),$$

and its *expected loss* over distribution P as

$$\mathcal{R}(\boldsymbol{w}) = \mathbb{E}_{\boldsymbol{z},\boldsymbol{z}',\boldsymbol{z}'' \sim P} L(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{z}', \boldsymbol{z}'').$$

The following theorem bounds the deviation between the empirical loss of the learned metric and its expected loss.

Theorem 1. *Let $\boldsymbol{w}^*$ be the optimal solution to SCML-Global with K basis elements, $\beta > 0$ and C constructed from $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ as above. Let $K^* \leq K$ be the number of nonzero entries in $\boldsymbol{w}^*$. Let us assume the norm of any instance bounded by some constant R and L uniformly upper-bounded by some constant U. Then for any $\delta > 0$, with probability at least $1 - \delta$ we have:*

$$\left|\mathcal{R}(\boldsymbol{w}^*) - \mathcal{R}^S_{emp}(\boldsymbol{w}^*)\right| \leq \frac{16\gamma R K^*}{\beta} + 3U\sqrt{\frac{N \ln 2 + \ln \frac{1}{\delta}}{0.5n}},$$

where N is the size of an γ-cover of $\mathcal{Z}$.

This bound has a standard $O(1/\sqrt{n})$ asymptotic convergence rate.[3] Its main originality is that it provides a theoretical justification to enforcing sparsity in our formulation. Indeed, notice that K^* (and not K) appears in the bound as a penalization term, which suggests that one may use a large basis set K without overfitting as long as K^* remains small. This will be confirmed by our experiments. A similar bound can be derived for mt-SCML, but not for SCML-Local because of its nonconvexity. Due to the lack of space, details and proofs can be found in the technical report version of this paper (Shi, Bellet, and Sha 2014).

Related Work

Global methods Most global metric learning methods learn the matrix $\boldsymbol{M}$ directly: see (Xing et al. 2002; Goldberger et al. 2004; Davis et al. 2007; Jain et al. 2008; Weinberger and Saul 2009) for representative papers. This is computationally expensive and subject to overfitting for moderate to high-dimensional problems. An exception is BoostML (Shen et al. 2012) which uses rank-one matrices as weak learners to learn a global Mahalanobis distance via a boosting procedure. However, it is not clear how BoostML can be generalized to multi-task or local metric learning.

Multi-task methods Multi-task metric learning was proposed in (Parameswaran and Weinberger 2010) as an extension to the popular LMNN (Weinberger and Saul 2009). The authors define the metric for task t as $d_t(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x} - \boldsymbol{x}')^{\mathrm{T}}(\boldsymbol{M}_0 + \boldsymbol{M}_t)(\boldsymbol{x} - \boldsymbol{x}')$, where $\boldsymbol{M}_t$ is task-specific and $\boldsymbol{M}_0$ is shared by all tasks. Note that it is straightforward to incorporate their approach in our framework by defining a shared weight vector $\boldsymbol{w}_0$ and task-specific weights $\boldsymbol{w}_t$.

Local methods MM-LMNN (Weinberger and Saul 2009) is an extension of LMNN which learns only a small number of metrics (typically one per class) in an effort to alleviate overfitting. However, no additional regularization is used and a full-rank metric is learned for each class, which becomes intractable when the number of classes is large. Like SCML-Local, PLML (Wang, Woznica, and Kalousis 2012) is based on a combination of metrics but there are major differences with our work: (i) weights only depend on a manifold assumption: they are not sparse and use no discriminative information, (ii) the basis metrics are full-rank, thus expensive to learn, and (iii) a weight vector is learned explicitly for each training instance, which can result in a large number of parameters and prevents generalization to new instances (in practice, for a test point, they use the weight vector of its nearest neighbor in the training set). Unlike the above discriminative approaches, GLML (Noh, Zhang, and Lee 2010) learns a metric independently for each point

[3] In robustness bounds, the cover radius γ can be made arbitrarily close to zero at the expense of increasing N. Since N appears in the second term, the right hand side of the bound indeed goes to zero when $n \to \infty$. This is in accordance with other similar learning bounds, for example, the original robustness-based bounds in (Xu and Mannor 2012).

Table 1: Datasets for global and local metric learning.

	Vehicle	Vowel	Segment	Letters	USPS	BBC
# samples	846	990	2,310	20,000	9,298	2,225
# classes	4	11	7	26	10	5
# features	18	10	19	16	256	9,636

in a generative way by minimizing the 1-NN expected error under some assumption for the class distributions.

Experiments

We first demonstrate the benefits of combining basis elements on global metric learning. Then we compare our methods to state-of-the-art algorithms on multi-task and local metric learning.[4] We use a 3-nearest neighbor classifier in all experiments. To generate a set of locally discriminative rank-one metrics, we first divide data into regions via clustering. For each region center, we select J nearest neighbors from each class (for $J = \{10, 20, 50\}$ to account for different scales), and apply Fisher discriminant analysis followed by eigenvalue decomposition to obtain the basis elements.[5]

Remark Due to the lack of space, some additional experimental results can be found in the technical report version of this paper (Shi, Bellet, and Sha 2014).

Global Metric Learning

Datasets We use 6 datasets from UCI[6] and BBC[7] (see Table 1). The dimensionality of USPS and BBC is reduced to 100 and 200 using PCA to speed up computation. We normalize the data as in (Wang, Woznica, and Kalousis 2012) and split into train/validation/test (60%/20%/20%), except for Letters and USPS where we use 3,000/1,000/1,000. Results are averaged over 20 random splits.

Setup Global metric learning is a convenient setting to study the effect of combining basis elements. To this end, we consider a formulation with the same loss function as SCML-Global but that directly learns the metric matrix with Frobenius norm regularization to reduce overfitting. We refer to it as Global-Frob. We generate the training triplets by identifying 3 target neighbors (nearest neighbors with same label) and 10 imposters (nearest neighbors with different label) for each instance. We tune the regularization parameter on the validation data. For SCML-Global, we use a basis set of 400 elements for Vehicle, Vowel, Segment and BBC, and 1,000 elements for Letters and USPS.

Results Table 2 shows misclassification rates with standard errors, where Euc is the Euclidean distance. The results show that SCML-Global performs similarly as Global-Frob on low-dimensional datasets but has *a clear advantage when dimensionality is high* (USPS and BBC). This demonstrates that learning a sparse combination of basis elements is an effective way to reduce overfitting and improve generalization. SCML-Global is also faster to train than Global-Frob on these datasets (about 2x faster on USPS and 3x on BBC) because it does not require PSD projections.

Table 2: Global metric learning results (best in bold).

Dataset	Euc	Global-Frob	SCML-Global
Vehicle	29.7±0.6	**21.5±0.8**	**21.3±0.6**
Vowel	**11.1±0.4**	**10.3±0.4**	**10.9±0.5**
Segment	5.2±0.2	**4.1±0.2**	**4.1±0.2**
Letters	14.0±0.2	**9.0±0.2**	**9.0±0.2**
USPS	10.3±0.2	5.1±0.2	**4.1±0.1**
BBC	8.8±0.3	5.5±0.3	**3.9±0.2**

Multi-task Metric Learning

Dataset Sentiment Analysis (Blitzer, Dredze, and Pereira 2007) is a popular dataset for multi-task learning that consists of Amazon reviews on four product types: kitchen appliances, DVDs, books and electronics. Each product type is treated as a task and has 1,000 positive and 1,000 negative reviews. To reduce computational cost, we represent each review by a 200-dimensional feature vector by selecting top 200 words of the largest mutual information with the labels. We randomly split the dataset into training (800 samples), validation (400 samples) and testing (400 samples) sets.

Setup We compare the following metrics: st-Euc (Euclidean distance), st-LMNN and st-SCML (single-task LMNN and single-task SCML-Global, trained independently on each task), u-Euc (Euclidean trained on the union of the training data from all tasks), u-LMNN (LMNN on union), u-SCML (SCML-Global on union), multi-task LMNN (Parameswaran and Weinberger 2010) and finally our own multi-task method mt-SCML. We tune the regularization parameters in mt-LMNN, st-SCML, u-SCML and mt-SCML on validation sets. As in the previous experiment, the number of target neighbors and imposters for our methods are set to 3 and 10 respectively. We use a basis set of 400 elements for each task for st-SCML, the union of these (1,600) for mt-SCML, and 400 for u-SCML.

Results Table 3 shows the results averaged over 20 random splits. First, notice that u-LMNN and u-SCML obtain significantly higher error rates than st-LMNN and st-SCML respectively, which suggests that the dataset may violate mt-LMNN's assumption that all tasks share a similar metric. Indeed, mt-LMNN does not outperform st-LMNN significantly. On the other hand, mt-SCML performs better than its single-task counterpart and than all other compared methods by a significant margin, demonstrating its ability to leverage some commonalities between tasks that mt-LMNN is unable to capture. It is worth noting that the solution found by mt-SCML is based on only 273 basis elements on average (out of a total of 1,600), while st-SCML makes use of significantly more elements (347 elements *per task* on average). Basis elements selected by mt-SCML are evenly distributed across all tasks, which indicates that it is able to exploit meaningful information across tasks to get both more accurate and more compact metrics. Finally, note that our algorithms are about an order of magnitude faster.

[4] For all compared methods we use MATLAB code from the authors' website. The MATLAB code for our methods is available at `http://www-bcf.usc.edu/~bellet/`.

[5] We also experimented with a basis set based on local GLML metrics. Preliminary results were comparable to those obtained with the procedure above.

[6] `http://archive.ics.uci.edu/ml/`

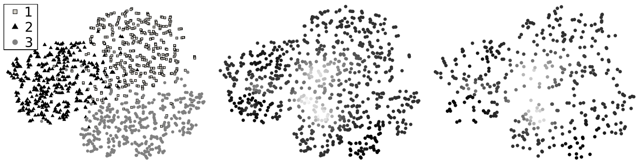
[7] `http://mlg.ucd.ie/datasets/bbc.html`

Table 3: Multi-task metric learning results.

Task	st-Euc	st-LMNN	st-SCML	u-Euc	u-LMNN	u-SCML	mt-LMNN	mt-SCML
Books	33.5±0.5	29.7±0.4	27.0±0.5	33.7±0.5	29.6±0.4	28.0±0.4	29.1±0.4	25.8±0.4
DVD	33.9±0.5	29.4±0.5	26.8±0.4	33.9±0.5	29.4±0.5	27.9±0.5	29.5±0.5	26.5±0.5
Electronics	26.2±0.4	23.3±0.4	21.1±0.5	29.1±0.5	25.1±0.4	22.9±0.4	22.5±0.4	20.2±0.5
Kitchen	26.2±0.6	21.2±0.5	19.0±0.4	27.7±0.5	23.5±0.3	21.9±0.5	22.1±0.5	19.0±0.4
Avg. accuracy	30.0±0.2	25.9±0.2	23.5±0.2	31.1±0.3	26.9±0.2	25.2±0.2	25.8±0.2	**22.9±0.2**
Avg. runtime	N/A	57 min	3 min	N/A	44 min	2 min	41 min	5 min

Table 4: Local metric learning results (best in bold).

Dataset	MM-LMNN	GLML	PLML	SCML-Local
Vehicle	23.1±0.6	23.4±0.6	22.8±0.7	**18.0±0.6**
Vowel	6.8±0.3	**4.1±0.4**	8.3±0.4	6.1±0.4
Segment	**3.6±0.2**	**3.9±0.2**	**3.9±0.2**	**3.6±0.2**
Letters	9.4±0.3	10.3±0.3	**8.3±0.2**	**8.3±0.2**
USPS	**4.2±0.7**	7.8±0.2	4.1±0.1	**3.6±0.1**
BBC	4.9±0.4	5.7±0.3	**4.3±0.2**	**4.1±0.2**
Avg. rank	2.0	2.7	2.0	**1.2**



(a) Class membership (b) Trained metrics (c) Test metrics

Figure 2: Illustrative experiment on 3 digits of USPS in 2D. A color version can be found in (Shi, Bellet, and Sha 2014).

Local Metric Learning

Setup We use the same datasets and preprocessing as for global metric learning. We compare SCML-Local to MM-LMNN (Weinberger and Saul 2009), GLML (Noh, Zhang, and Lee 2010) and PLML (Wang, Woznica, and Kalousis 2012). The parameters of all methods are tuned on validation sets or set by authors' recommendation. MM-LMNN use 3 target neighbors and all imposters, while these are set to 3 and 10 in PLML and SCML-Local. The number of anchor points in PLML is set to 20 as done by the authors. For SCML-Local, we use the same basis set as SCML-Global, and embedding dimension D' is set to 40 for Vehicle, Vowel, Segment and BBC, and 100 for Letters and USPS.

Results Table 4 gives the error rates along with the average rank of each method across all datasets. Note that SCML-Local significantly improves upon SCML-Global on all but one dataset and achieves the best average rank. PLML does not perform well on small datasets (Vehicle and Vowel), presumably because there are not enough points to get a good estimation of the data manifold. GLML is fast but has rather poor performance on most datasets because its Gaussian assumption is restrictive and it learns the local metrics independently. Among discriminative methods, SCML-Local offers the best training time, especially for high-dimensional data (e.g. on BBC, it is about 5x faster than MM-LMNN and 15x faster than PLML). Note that on this dataset, both MM-LMNN and PLML perform worse than SCML-Global due to severe overfitting, while SCML-Local avoids it by learning significantly fewer parameters.

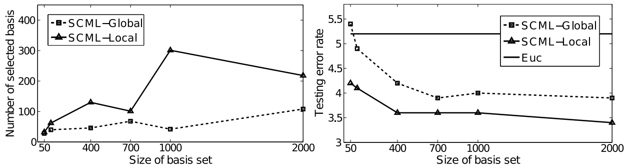


Figure 3: Effect of the number of bases on Segment dataset.

Visualization of the learned metrics To provide a better understanding of why SCML-Local works well, we apply it to digits 1, 2, and 3 of USPS projected in 2D using t-SNE (van der Maaten and Hinton 2008), shown in Figure 2(a). We use 10 basis elements and $D' = 5$. Figure 2(b) shows the training points colored by their learned metric (based on the projection of the weight vectors in 1D using PCA). We see that the local metrics vary smoothly and are thereby robust to outliers. Unlike MM-LMNN, points within a class are allowed to have different metrics: in particular, this is useful for points that are near the decision boundary. While smooth, the variation in the weights is thus driven by discriminative information, unlike PLML where they are only based on the smoothness assumption. Finally, Figure 2(c) shows that the metrics consistently generalize to test data.

Effect of the basis set size Figure 3 shows the number of selected basis elements and test error rate for SCML-Global and SCML-Local as a function of the size of basis set on Segment (results were consistent on other datasets). The left pane shows that the number of selected elements increases sublinearly and eventually converges, while the right pane shows that test error may be further reduced by using a larger basis set without significant overfitting, as suggested by our generalization bound (Theorem 1). Figure 3 also shows that SCML-Local generally selects more basis elements than SCML-Global, but notice that it can outperform SCML-Global even when the basis set is very small.

Conclusion

We proposed to learn metrics as sparse combinations of rank-one basis elements. This framework unifies several paradigms in metric learning, including global, local and multi-task learning. Of particular interest is our local metric learning algorithm which can compute instance-specific metrics for both training and test points in a principled way. The soundness of our approach is supported theoretically by a generalization bound, and we showed in experimental studies that the proposed methods improve upon state-of-the-art algorithms in terms of accuracy and scalability.

Acknowledgements

This research is partially supported by the IARPA via DoD/ARL contract # W911NF-12-C-0012 and DARPA via contract # D11AP00278. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, DARPA, or the U.S. Government.

References

Argyriou, A.; Evgeniou, T.; and Pontil, M. 2008. Convex multi-task feature learning. *Mach. Learn.* 73(3):243–272.

Bellet, A., and Habrard, A. 2012. Robustness and Generalization for Metric Learning. Technical report, arXiv:1209.1086.

Bellet, A.; Habrard, A.; and Sebban, M. 2013. A Survey on Metric Learning for Feature Vectors and Structured Data. Technical report, arXiv:1306.6709.

Blitzer, J.; Dredze, M.; and Pereira, F. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *ACL*.

Caruana, R. 1997. Multitask Learning. *Mach. Learn.* 28(1):41–75.

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

Duchi, J., and Singer, Y. 2009. Efficient Online and Batch Learning Using Forward Backward Splitting. *JMLR* 10:2899–2934.

Frome, A.; Singer, Y.; Sha, F.; and Malik, J. 2007. Learning Globally-Consistent Local Distance Functions for Shape-Based Image Retrieval and Classification. In *ICCV*.

Goldberger, J.; Roweis, S.; Hinton, G.; and Salakhutdinov, R. 2004. Neighbourhood Components Analysis. In *NIPS*.

Gong, P.; Ye, J.; and Zhang, C. 2012. Robust multi-task feature learning. In *KDD*.

Hauberg, S.; Freifeld, O.; and Black, M. 2012. A Geometric take on Metric Learning. In *NIPS*.

Hong, Y.; Li, Q.; Jiang, J.; and Tu, Z. 2011. Learning a mixture of sparse distance metrics for classification and dimensionality reduction. In *CVPR*.

Jain, P.; Kulis, B.; Dhillon, I.; and Grauman, K. 2008. Online Metric Learning and Fast Similarity Search. In *NIPS*.

Kulis, B. 2012. Metric Learning: A Survey. *Foundations and Trends in Machine Learning* 5(4):287–364.

Noh, Y.-K.; Zhang, B.-T.; and Lee, D. 2010. Generative Local Metric Learning for Nearest Neighbor Classification. In *NIPS*.

Parameswaran, S., and Weinberger, K. 2010. Large Margin Multi-Task Metric Learning. In *NIPS*.

Ramanan, D., and Baker, S. 2011. Local Distance Functions: A Taxonomy, New Algorithms, and an Evaluation. *TPAMI* 33(4):794–806.

Schölkopf, B.; Smola, A.; and Müller, K.-R. 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.* 10(1):1299–1319.

Shen, C.; Kim, J.; Wang, L.; and van den Hengel, A. 2012. Positive Semidefinite Metric Learning Using Boosting-like Algorithms. *JMLR* 13:1007–1036.

Shi, Y.; Bellet, A.; and Sha, F. 2014. Sparse Compositional Metric Learning. Technical report, arXiv:1404.4105. http://arxiv.org/pdf/1404.4105.

van der Maaten, L., and Hinton, G. 2008. Visualizing Data using t-SNE. *JMLR* 9:2579–2605.

Wang, J.; Woznica, A.; and Kalousis, A. 2012. Parametric Local Metric Learning for Nearest Neighbor Classification. In *NIPS*.

Weinberger, K., and Saul, L. 2009. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *JMLR* 10:207–244.

Xiao, L. 2010. Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization. *JMLR* 11:2543–2596.

Xing, E.; Ng, A.; Jordan, M.; and Russell, S. 2002. Distance Metric Learning with Application to Clustering with Side-Information. In *NIPS*.

Xu, H., and Mannor, S. 2012. Robustness and Generalization. *Mach. Learn.* 86(3):391–423.

Yang, X.; Kim, S.; and Xing, E. 2009. Heterogeneous multitask learning with joint sparsity constraints. In *NIPS*.

Ying, Y., and Li, P. 2012. Distance Metric Learning with Eigenvalue Optimization. *JMLR* 13:1–26.

Yuan, M., and Lin, Y. 2006. Model selection and estimation in regression with grouped variables. *J. Roy. Statist. Soc. Ser. B* 68:49–67.

Zhan, D.-C.; Li, M.; Li, Y.-F.; and Zhou, Z.-H. 2009. Learning instance specific distances using metric propagation. In *ICML*.