Geometry-aware Metric Learning

Zhengdong Lu LUZ@CS.UTEXAS.EDU
Prateek Jain PJAIN@CS.UTEXAS.EDU
Inderjit S. Dhillon INDERJIT@CS.UTEXAS.EDU
University of Texas at Austin, Austin, TX 78712

Abstract

In this paper, we introduce a generic framework for semi-supervised kernel learning. Given pairwise (dis-)similarity constraints, we learn a kernel matrix over the data that respects the provided side-information as well as the local geometry of the data. Our framework is based on metric learning methods, where we jointly model the metric/kernel over the data along with the underlying manifold. Furthermore, we show that for some important parameterized forms of the underlying manifold model, we can estimate the model parameters and the kernel matrix efficiently. Our resulting algorithm is able to incorporate local geometry into the metric learning task; at the same time it can handle a wide class of constraints. Finally, our algorithm is fast and scalable – unlike most of the existing methods, it is able to exploit the low dimensional manifold structure and does not require semi-definite programming. We demonstrate wide applicability and effectiveness of our framework by applying to various machine learning tasks such as semi-supervised classification, colored dimensionality reduction, manifold alignment etc. On each of the tasks our method performs competitively or better than the respective state-of-the-art method.

1. Introduction

Over the years, kernel methods have become an important tool in many machine learning tasks. Success of these methods is critically dependent on selecting an appropriate kernel for the given task at hand using the provided side-information. To this end, there have been several recent approaches to learn a kernel function, *e.g.*, (Zhu et al., 2005; Lanckriet et al., 2004; Davis et al., 2007; Ong et al., 2005). In most real-life applications, the volume of available data is huge but the amount of supervision available is very limited. This necessitates semi-supervised methods for kernel learning that also exploit the geometry of the data. Additionally, the side-information can be provided in a variety of forms, *e.g.*, labels, (dis-)similarity constraints, and click through feedback. Thus, a generic framework is required for semi-supervised kernel learning that is able to handle different types of supervision, while exploiting the intrinsic structure of the unsupervised data. Furthermore, the learning algorithm should be fast and scalable to handle large volumes of data. While existing kernel learning algorithms have been shown to perform well across various applications, most fail to satisfy some of these basic requirements.

In this paper, we propose a framework for semi-supervised kernel learning that is based on a generalization of existing work on metric learning (Davis et al., 2007; Weinberger et al., 2006; Globerson & Roweis, 2005), as well as data-dependent kernels (Zhu et al., 2005; Sindhwani et al., 2005). Metric learning provides a flexible method to learn a task-dependent distance function (kernel) over the data points using the provided distance constraints. However, it is critically dependent on the existing feature representation or a pre-defined similarity function, and does not take into account the manifold structure of the provided data. On the other hand, data-dependent kernel learning approaches exploit the intrinsic structure of the provided data, but typically do not specialize to a given task.

Our framework incorporates the intrinsic structure in the data, while learning a task-dependent kernel. Specifically, we jointly model a task-dependent kernel as well as a data-dependent kernel that reflects the local geometry or manifold structure of the data. We show that for some important parameterizations of the set of data-dependent kernels, our formulation admits convexity, and the proposed optimization algorithm efficiently learns an appropriate kernel function for the given task. Our algorithm is fast, scalable, does not involve semi-definite programming, and crucially, is able to exploit the low dimensional structure of the underlying manifold that is often present in real-world datasets.

Our proposed framework is generic and can be easily tai-

Appearing in *Proceedings of the 26^{th} International Conference on Machine Learning*, Montreal, Canada, 2009.

lored for a variety of tasks. In this paper, we apply our method to the task of classification (inductive and transductive setting), automatic model selection for standard kernel functions, and semi-supervised manifold learning. For each application, we empirically demonstrate that our method can achieve comparable or better performance than the respective state-of-the-art.

2. Previous Work

Existing kernel learning methods can be broadly divided into two categories. The first category includes primarily task-dependent approaches, where the intrinsic structure in the data is assumed, and the goal is to maximally tune the kernel to the provided side-information for the given task, *e.g.*, class labels for classification, must (cannot)-link constraints for semi-supervised clustering. Prominent methods include metric learning (Davis et al., 2007), multiple kernel learning (Lanckriet et al., 2004; Sonnenburg et al., 2006), hyper-kernels (Ong et al., 2005), hyper-parameter cross validation (Seeger, 2008), etc.

The other category of kernel learning methods consist of data-dependent approaches, which explicitly model the geometry of the data, *e.g.*, underlying *manifold* structure. These methods appear in both unsupervised and semi-supervised learning scenarios. For the unsupervised case, (Weinberger et al., 2004) proposed a method to recover the underlying low dimensional manifold by learning a kernel over it. More generally, (Bengio et al., 2004) show that a large class of manifold learning methods are equivalent to learning certain types of kernels. For the semi-supervised setting, data-dependent kernels are used to enforce smoothness on a graph or a similar structure composed from *all* of the data. Like in the unsupervised case, the kernel captures the manifold and/or cluster structure of the data, and after integrated a regularized classification model, often provides good generalization performance (Sindhwani et al., 2005; Chapelle et al., 2003; Seeger, 2002).

Our proposed method combines the two kernel learning paradigms, thereby exploiting the geometry of the data while retaining the task-specific feature. Related work in this direction is limited and largely focuses on learning parameters for a specific family of data-dependent kernels, *e.g.*, spectral kernels (Zhu et al., 2005; Sun et al., 2008; Lafferty & Lebanon, 2005). In comparison, our method is based on a non-parametric information-theoretic metric/kernel learning method and is more flexible. Furthermore, existing methods are typically designed for a particular application only, *e.g.*, semi-supervised classification, and are not able to handle different type of constraints, such as distance constraints. In contrast, our proposed framework can handle a variety of constraints and is applicable to various machine learning tasks (see Section 6).

3. Methodology

Given a set of n points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\} \in \mathbb{R}^d$, we seek a positive semi-definite kernel matrix K that can be later used for various tasks, *e.g.* classification, retrieval, etc. Our goal is two-fold: 1) use the provided supervision over the data, 2) exploit the unlabeled or unsupervised data, i.e., we want to learn a kernel that respects the underlying manifold structure in the data while also incorporating the side-information provided. Previous kernel learning approaches typically handle this problem by learning a spectral kernel $K = \sum_i \alpha_i \boldsymbol{v}_i \boldsymbol{v}_i^T$, where the vectors $\boldsymbol{v}_i$ are the low-frequency eigenvectors of the Laplacian of a k-NN graph. However, constraining the eigenvectors to be unchanged severely restricts the class of kernels that can be learned.

A contrasting task-dependent approach to kernel learning is based on the metric learning paradigm, where the goal is to learn a kernel K that is "close" to a pre-defined baseline kernel K_0 and satisfies the provided pairwise (or relative) constraints that are specific to the task at hand. Formally, K is obtained by solving the following problem:

$$\min_K \; D(K, K_0), \quad \text{s.t.} \quad K \in \mathcal{K},$$

where $\mathcal{K}$ is a convex set of kernel K that satisfy

$$\begin{aligned} & K_{ii} + K_{jj} - 2K_{ij} \leq u \;\; (i,j) \in \mathcal{S}, \\ & K_{ii} + K_{jj} - 2K_{ij} \geq l \;\; (i,j) \in \mathcal{D}, \\ & K \succeq 0. \end{aligned} \tag{1}$$

In the above $\mathcal{S}$ is the given set of similar points, $\mathcal{D}$ is the given set of dis-similar points, and $D(\cdot,\cdot)$ is a distance function for comparing two kernel matrices. We will denote the set of kernel that satisfy (1) as the set of task-dependent kernel. Although flexible and effective for various problems, this framework does not account for the unlabeled data and their geometry. As a result, large amount of supervision is required to capture the intrinsic structure in the data.

In this paper, we propose a geometry-aware metric learning (G-ML) framework that combines both the data-dependent and task-dependent kernel learning approaches. Our model maintains the flexibility of the metric learning based approach while exploiting the intrinsic structure in the data, and as we shall show later, engenders multiple competitive machine learning models.

3.1. Geometry-aware Metric Learning

In this section, we describe our geometry-aware metric learning (G-ML) model, where we learn the kernel K, as well as the kernel M that explicitly exploits the intrinsic structure in the data through the optimization problem:

$$\min_{K,M} \; D(K, M), \quad \text{s.t.} \quad K \in \mathcal{K},\; M \in \mathcal{M}, \tag{2}$$

where the set $\mathcal{M}$ is a parametric set of kernels that capture the intrinsic geometry of the labeled as well as unlabeled data and $D(\cdot,\cdot)$ is a distance function over matrices. The above optimization problem computes kernel K that satisfies task specific constraints (1) and is also close to kernel M, thus incorporating data geometry into the kernel K (see Figure 1). Later in the section, we give a few interesting examples of the set $\mathcal{M}$.

A key component of our framework is the distance function $D(K, M)$ that is being used. In this work, we use the LogDet matrix divergence, $D_{\ell d}(K, M)$, as the distance function, where $D_{\ell d}(K, M) = \operatorname{tr}(KM^{-1}) - \log\det(KM^{-1}) - n$. The LogDet divergence is a Bregman matrix divergence that is typically defined over positive definite matrices and its definition can be extended to the case when the range space of matrix K is the same as that of M. Previously, (Davis et al., 2007) showed the efficacy of LogDet as a matrix divergence for kernel learning, and pointed out its equivalence to metric learning (called information-theoretic metric learning, or ITML).

Now, we give an important example of the set $\mathcal{M}$ based on spectral learning methods (Zhu et al., 2005) that captures the underlying structure in the data. First, define a graph $\mathcal{G}$ over the data points that captures local structure of data, *e.g.*, a k-NN graph or an ϵ-ball graph. Let W be the adjacency matrix of $\mathcal{G}$, D be the degree matrix, L be the graph Laplacian[1] $L = D - W$, and $V = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_r]$ be the r eigenvectors of L (typically $r \ll n$) corresponding to the smallest eigenvalues of L: $\lambda_1 \leq \cdots \leq \lambda_r$. Then, the set $\mathcal{M}$ we consider is given by:

$$\mathcal{M} = \left\{ \sum_i^r \alpha_i \boldsymbol{v}_i \boldsymbol{v}_i^T \,|\, \alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_r \geq 0. \right\} \tag{3}$$

where the order constraints $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_r \geq 0$ further ensure smoothness (the eigenvector $\boldsymbol{v}_i$ is known to be smoother than $\boldsymbol{v}_{i+1}$).

For this particular choice of $\mathcal{M}$, the kernel K is obtained by solving the following optimization problem:

$$\begin{aligned} \min_{K,\alpha_1,\alpha_2,\ldots,\alpha_r} \quad & D_{\ell d}(K, M) \\ \text{s.t. } & K \in \mathcal{K},\ M = \sum_i^r \alpha_i \boldsymbol{v}_i \boldsymbol{v}_i^T, \\ & \alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_r \geq 0. \end{aligned} \tag{4}$$

Solving above problem yields $\{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ in the cone $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_r \geq 0$ and a feasible kernel K that is close to $M = \sum_i^r \alpha_i \boldsymbol{v}_i \boldsymbol{v}_i^T$ (see Figure 1). Slack variables can be incorporated in our framework to ensure that the set $\mathcal{K}$ is always feasible, even under noisy constraints.

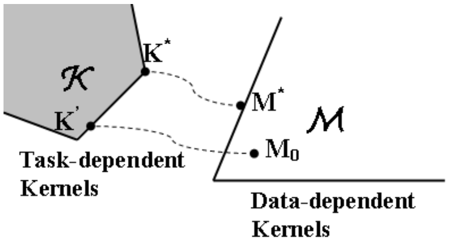

Figure 1. Illustration of G-ML. The shadowed polygon stands for the feasible set of kernels K specified by the task dependent pairwise constraints. The cone stands for data-dependent kernels that exploits the intrinsic geometry of the data. Using a fixed M_0 would lead to sub-optimal kernel K', while the joint optimization (as in (2)) over both M and K leads to a better solution K^*.

3.2. Alternative $\mathcal{M}$

In the above subsection, we discussed an example of $\mathcal{M}$ as a particular subset of spectral kernels. However our framework is general and depending on the application it can admit other parametric sets also. For example, consider the set:

$$\mathcal{M} = \{S - S(I + TS)^{-1}TS |\, T = \sum_i^r \theta_i \boldsymbol{v}_i \boldsymbol{v}_i^T, \quad \theta_1 \geq \cdots \geq \theta_r \geq 0.\} \tag{5}$$

where S is a fixed given kernel and the vectors $\boldsymbol{v}_i$ are eigenvectors of the graph Laplacian L. This set generalizes the data-dependent kernel proposed by (Sindhwani et al., 2005) by replacing the graph Laplacian with a more flexible T. Note that $\mathcal{M}$ given by (5) reduces to $\mathcal{M}$ given by (3) in the limit $\|S^{-1}\| \to 0$. This set of kernel is interesting in that, unlike most spectral kernels that are usually evaluated in a transductive setting, the kernel value can be naturally extended to unseen samples as

$$M(\boldsymbol{x}, \boldsymbol{x}') = S(\boldsymbol{x}, \boldsymbol{x}') - S(\boldsymbol{x}, .)(I + TS)^{-1}TS(., \boldsymbol{x}')$$

As will be shown in Section 4, the set $\mathcal{M}$ given by (3) as well as (5) both lead to convex sub-problems for finding T with fixed K. In general, the convexity holds if $\{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_r\}$ are orthogonal, which allows us to extend our model to other manifold learning models (Bengio et al., 2004), such as Isomap or LLE. The set $\mathcal{M}$ can also be adapted to perform automatic model selection for *supervised learning*, for example we can tune the parameter for the RBF kernels by letting

$$\mathcal{M} = \left\{ \alpha \exp(-\frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}{2\sigma^2}) \mid \alpha > 0, \sigma > 0 \right\}, \tag{6}$$

where α and σ are parameters to be learned by G-ML.

4. Algorithm

In this section, we analyze properties of the proposed optimization problem (4) and propose a fast and scalable algorithm. First, note that although the constraints specified in (4) are all linear, the objective function $D_{\ell d}(K, M)$ is not jointly convex in K and M. However, the problem can be shown to be convex individually in K and M^{-1}. Here and in the remainder of the paper, whenever the inverse of a matrix does not exist, we use its Moore-Penrose inverse.

[1] We can also use the normalized Laplacian $I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$.

Algorithm 1 Geometry-aware Metric Learning(G-ML) Optimization procedure when $\mathcal{M}$ is given by (3)

```
Input: $X$: input $d \times n$ matrix, $\mathcal{S}$: similarity constraints
       $\mathcal{D}$: dis-similarity constraints, $\boldsymbol{\alpha}^0$: initial $\boldsymbol{\alpha}$
       $\gamma$: slack parameter, $r$: number of eigenvectors
Output: $K, M$
$\mathcal{G}$ =kNN-graph($X$), $W$ =Affinity matrix of $\mathcal{G}$
$L = D - W, L = \sum_i^n \mu_i \boldsymbol{v}_i \boldsymbol{v}_i^T$
$M = \sum_i^r \alpha_i^0 \boldsymbol{v}_i \boldsymbol{v}_i^T$
repeat
    $K =$ ITML($M, \mathcal{S}, \mathcal{D}, \gamma$) //(Step A)
    $\boldsymbol{\alpha} =$ FindAlpha($K, \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_r$) //(Step B)
    $M = \sum_i \alpha_i \boldsymbol{v}_i \boldsymbol{v}_i^T$
until convergence
```

```
function $\boldsymbol{\alpha} =$ FindAlpha($K, \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_r$)
Cyclic projection method to solve (4) with fixed $K$
$\alpha_i = \boldsymbol{v}_i^T K \boldsymbol{v}_i, 1 \leq i \leq r$
$\boldsymbol{\nu} = 0, i = 0$
repeat
    $c = \min(\nu_i, (\alpha_{i+1} - \alpha_i)/2)$
    $\nu_i = \nu_i - c, \alpha_{i+1} = \alpha_{i+1} - c, \alpha_i = \alpha_i + c$
    $i = \mod (i+1, n)$
until convergence
```

It is easy to see that on fixing M, the problem is strictly convex in K as $D_{\ell d}(K, M)$ is known to be convex in K (Davis et al., 2007). The following lemma shows that (4) is also convex in the parameters $\frac{1}{\alpha_i}, 1 \leq i \leq r$, when K is fixed.

Lemma 1. *Assuming K to be fixed, Problem* (4) *is convex in $\beta_1 = 1/\alpha_1, \beta_2 = 1/\alpha_2, \ldots, \beta_r = 1/\alpha_r$.*

Proof. Since $M^{-1} = \sum_i \beta_i \boldsymbol{v}_i \boldsymbol{v}_i^T$, where $\beta_i = \frac{1}{\alpha_i}$, the fact that $D_{\ell d}(K, M) = D_{\ell d}(M^{-1}, K^{-1})$ is convex in M^{-1} implies convexity in $\beta_i, \forall i$. Furthermore, the constraints $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_r \geq 0$ can be equivalently written as a set of linear constraints $\beta_r \geq \cdots \geq \beta_2 \geq \beta_1 > 0$. □

Now, we describe our proposed alternating minimization algorithm for solving (4). Our algorithm is based on individual convexity of (4) w.r.t K and M^{-1}. It iterates by fixing M (or equivalently $\alpha_1, \alpha_2, \ldots, \alpha_r$) to solve for K (denoted Step A), and then fixing K to solve for $\alpha_1, \alpha_2, \ldots, \alpha_r$ (Step B). In Step A, to find K, we use the cyclic projection algorithm where at each step we project the current solution onto one of the constraints. The projection problem that needs to be solved at each step is:

$$\min_{K} D_{\ell d}(K, K_t), \quad \text{s.t. } K_{ii} + K_{jj} - 2K_{ij} \leq u,$$

i.e., projection w.r.t. single (dis-)similarity constraint. As shown in (Davis et al., 2007), the above problem can be solved in closed form using a one-rank update to K_t. Furthermore, the update can be computed in just $\mathcal{O}(nk)$ operations, where $r \ll n$ is the rank of the kernel M. Now in Step B, to obtain $\alpha_1, \alpha_2, \ldots, \alpha_r$, we solve the equivalent optimization problem:

$$\min_{\beta_1, \beta_2, \ldots, \beta_r} D_{\ell d}(\sum_i \beta_i \boldsymbol{v}_i \boldsymbol{v}_i^T, K^{-1})$$
$$\text{s.t. } \beta_r \geq \beta_{r-1} \geq \cdots \geq \beta_1 \geq 0,$$

where $\beta_i = 1/\alpha_i$. This problem can also be solved using cyclic projection, where at each step the current solution is projected onto one of the inequality constraints. Every projection step can be performed in just $\mathcal{O}(k)$ operations.

In summary, we have presented a highly scalable and easy to implement algorithm (Algorithm 1) for solving (4). Furthermore, the objective function value achieved by our algorithm is guaranteed to converge.

Alternative $\mathcal{M}$ As mentioned in Section 3.2, an alternate set $\mathcal{M}$ given by (5) induces an natural out-of-sample extension. Although it is not further pursued in this paper, we would like to point out that, similar to (4), this alternative set $\mathcal{M}$ also leads to a convex optimization problem for computing M when K is fixed.

Lemma 2. *Assuming K to be fixed, Problem* (4) *is convex in $\theta_1, \theta_2, \ldots, \theta_r$.*

Proof. Restricting the kernel function to the provided samples, we get $M = S - S(I + TS)^{-1}TS$. Using the Sherman-Morrison-Woodbury formula, $M^{-1} = S^{-1} + T$. Now, $D_{\ell d}(K, M)$ is convex in M^{-1}. Using the property that a function $g(x) = f(a + x)$ is convex if f is convex, $D_{\ell d}(K, M)$ is convex in T. As T is a linear function of $\theta_i, 1 \leq i \leq r$, $D_{\ell d}(K, M)$ is convex in $\theta_1, \cdots, \theta_r$. □

Using the above lemma, we can adapt Algorithm 1 to obtain a suboptimal solution to (2) where $\mathcal{M}$ is given by (5).

Unlike the kernels in (3) and (5), the set $\mathcal{M}$ given by (6) does not admit a convex subprolem when fixing K. However, since only two parameters are involved, we can still adapt our alternative minimization framework to obtain a reasonably efficient method for optimizing (2) using $\mathcal{M}$ specified in (6).

5. Discussion

5.1. Connection to Regularization Theory

Now, we present a regularization theory based interpretation of our methodology for estimating kernel K (Problem (4)). Using duality theory, it can be shown that the general form of the solution to (4) is given by:

$$K = (\sum_i \alpha_i^{-1} \boldsymbol{v}_i \boldsymbol{v}_i^T + \sum_{(i,j)\in\mathcal{S}} \gamma_{ij}^{\mathcal{S}} (\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^T - \sum_{(i,j)\in\mathcal{D}} \gamma_{ij}^{\mathcal{D}} (\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^T)^{-1} \quad (7)$$

with $\gamma_{ij}^{\mathcal{S}}, \gamma_{ij}^{\mathcal{D}} \geq 0$ and $\boldsymbol{e}_i$ being the vector with the i^{th} entry one and rest zeros. Let $f : X \to \mathbb{R}$ be a real valued function over the feature space and $\boldsymbol{f} = [f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \cdots, f(\boldsymbol{x}_n)]^T$, we then have

$$\boldsymbol{f}^T K^{-1} \boldsymbol{f} = \boldsymbol{f}^T \left(\sum_i \frac{1}{\alpha_i} \boldsymbol{v}_i \boldsymbol{v}_i^T \right) \boldsymbol{f} + \sum_{(i,j) \in \mathcal{S}} \gamma_{ij}^{\mathcal{S}} (f_i - f_j)^2 - \sum_{(i,j) \in \mathcal{D}} \gamma_{ij}^{\mathcal{D}} (f_i - f_j)^2 \quad (8)$$

where the first term addresses the overall smoothness of function f on the graph, while the last two terms measures the violation of pairwise constraints. Formulation (8) generalizes the joint regularization framework proposed by (Sindhwani et al., 2005) to include non-positive definite term (dis-similarity term $\sum_{(i,j)\in\mathcal{D}} \gamma_{ij}^{\mathcal{D}}(f_i - f_j)^2$ in our case) in the regularization, while the overall positive definiteness is still ensured either explicitly through another constraint ($K \succeq 0$) or implicitly through the particular optimization algorithm (Bregman projection in our case).

5.2. Connection to Gaussian Processes(GP)

Next, we present an interesting connection of our method to that of GP based methods for estimating M. Let $K = \Phi(X)\Phi(X)^T$, where $\Phi(X) = [\phi(\boldsymbol{x}_1)\, \phi(\boldsymbol{x}_2) \cdots \phi(\boldsymbol{x}_n)]^T$ and $\phi(\boldsymbol{x}_i) \in \mathbb{R}^m$ is the feature space representation of point $\boldsymbol{x}_i$. As in standard GP based methods, assume that each of the feature dimension of $\phi(\boldsymbol{x}_i)$'s are jointly Gaussian with mean 0 and covariance M that needs to be estimated. Thus, the likelihood of the data is given by:

$$\mathcal{L} = \frac{1}{(2\pi)^{n/2}|M|^{1/2}} \exp\left(-\frac{1}{2} \operatorname{tr}\left(\Phi(X)^T M^{-1} \Phi(X) \right) \right).$$

It is easy to see that maximizing the above given likelihood is equivalent to minimizing $D_{\ell d}(K, M)$ with fixed K. Assuming a parametric form for $M = \sum_i \alpha_i \boldsymbol{v}_i \boldsymbol{v}_i^T$, GP based spectral kernel learning is equivalent to learning M using our method. Furthermore, typical GP based methods use one-rank target alignment kernel $K = yy^T$, where y_i is the label of i-th point. In contrast, we use a more robust learned kernel K that not only accounts for the labels, but also the similarity in the data points itself, i.e. our learned kernel K is less likely to overfit to the provided labels and is applicable to a wider class of problems where supervision need not be in the form of labels.

6. Applications

In this section, we describe a few applications of our geometry-aware metric learning framework (G-ML) for kernel learning. Besides enhancing existing metric/kernel learning methods, our method also extends the application of kernel learning to a few previously inapplicable tasks as well. In particular, we will demonstrate the effectiveness of our framework on a variety of classification and manifold learning tasks.

6.1. Classification

First, we describe application of our method to the task of classification. Depending on the given problem, we have two scenarios: 1) supervised case where the test points are unknown in the training phase, and 2) semi-supervised case where the test/unlabeled points are also part of the training. For both the cases, pairwise similarity/dis-similarity constraints are obtained using the provided labels over the data, and the k nearest neighbor classifier with the learned kernel K is used for predicting the labels. In the supervised learning scenario, we apply G-ML to the task of automatic model selection by learning the parameters for the baseline kernel M. For semi-supervised learning, G-ML jointly learns the kernel K and the eigenvalues of the spectral kernel M, thereby taking into account the geometry of the unlabeled data. Note that the optimization step for M (step B) is similar to the kernel-target alignment technique for selecting a spectral kernel (Zhu et al., 2005). However, (Zhu et al., 2005) treat the kernel as a long vector, while our method respects the two-dimensional structure and positive definiteness of the matrix M.

6.2. Manifold Learning

G-ML is applicable to semi-supervised manifold learning where the task is to learn and exploit the underlying manifold structure using the provided supervision (pairwise (dis-)similarity constraints). In particular, we apply G-ML to the task of non-linear dimensionality reduction and manifold alignment. In contrast to other metric learning methods (Xing et al., 2002; Davis et al., 2007) that learns the metric over the ambient space, G-ML learns the metric *on the manifold*, where $\{\boldsymbol{v}_i\}$ are the approximate coordinates of the data on the manifold (Belkin & Niyogi, 2003).

Colored Dimensionality Reduction Here we consider the semi-supervised dimensionality reduction task where we want to retain both the intrinsic manifold structure of data and the (partial) label information. G-ML naturally merges the two sources of information; the learned kernel K incorporates the manifold structure (as expressed in $\{\alpha_i\}$ and $\{\boldsymbol{v}_i\}$) while reflecting the provided side information (expressed through constraints). Hence, the leading eigenvectors of K should provide a better low-dimensional representation of the data. In absence of any constraints, this dimension reduction model degenerates to Laplacian Eigenmaps (Belkin & Niyogi, 2003). Furthermore, compared to (Song et al., 2007), our model is able to learn a more accurate embedding of the data (Figure 4).

Manifold Alignment Finally, we apply our method to the task of manifold alignment, where the goal is to align previously disconnected (or weakly connected) manifolds according to some common property. For example, consider images of different objects under a particular transformation, *e.g.* rotation, illumination, scaling etc, which will form a low-dimensional manifold called Lie group. The

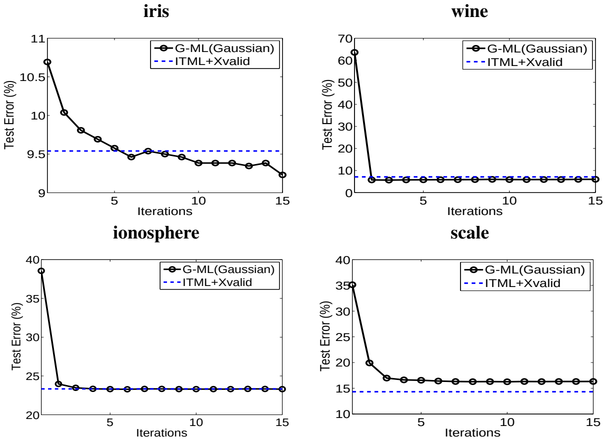

Figure 2. 4-NN classification error via kernels learned using our method (G-ML) and ITML (Davis et al., 2007). The data-dependent kernel M is the RBF kernel. Clearly, G-ML is able to achieve competitive error rate while learning the kernel width for M, while ITML requires cross validation.

goal is to estimate information about the *transformation* of the object in the image, rather than the object itself. We show that G-ML accurately represents the corresponding Lie group manifold by aligning the image manifold of different objects under the same transformation (captured by a joint graph Laplacian). This alignment is achieved through learning the kernel K by constraining a small subset of images with similar transformations to have small distance.

7. Experimental Results

In this section, we evaluate our method for geometry-aware metric learning (G-ML) on the applications mentioned in the previous section. Specifically, we apply our method to the task of classification, semi-supervised classification, non-linear dimensionality reduction, and manifold alignment. For each task we compare our method with the respective state-of-the-art methods.

7.1. Classification: Supervised Learning

First, we apply our G-ML framework to the task of classification in a supervised learning scenario (Section 6.1). For this task, we consider the feasible set $\mathcal{M}$ for M to be scaled Gaussian RBF kernels with unknown scale α and kernel width σ, as in (6). Unlike the spectral kernel case, the sub-problem for finding α and σ is non-convex and a local optimum for the non-convex subproblem is found with conjugate gradient descent (Matlab function fminsearch). The resulting K is then used for k-NN classification. We evaluate our method (G-ML) on four standard UCI datasets (iris, wine, balance-scale and ionosphere). For each dataset we use 20 points for training and the rest for testing. Figure 2 compares 4-NN classification error incurred by our method to that of the state-of-the-art ITML method (Davis et al., 2007). For ITML, the kernel width of Gaussian RBF M is selected using leave-one-out cross validation. Clearly, G-ML is able to automatically select a good kernel width, while ITML requires slower cross validation to obtain a similar width parameter.

7.2. Classification: Semi-supervised Learning

Next, we evaluate our method for classification in the semi-supervised setting (Section 6.1). We evaluate our method on four datasets that fall in two broad categories: a) text classification: two standard subsets of 20-newsgroup dataset, namely, baseball-hockey (1993 instances/ 2 classes), and pc-mac (1943/2). b) digit classification: two subsets of USPS digits dataset, odd-even (4000/2) and ten digits (4000/10). Odd-even involves classifying odd "1, 3, 5, 7, 9" vs. even "0, 2, 4, 6, 8" digits, while ten digits is the standard 10-way digit classification.

To form the k-NN graph, we use cosine similarity over tf-idf representation for text classification datasets and RBF-kernel function over gray-scale pixel values for the digits dataset. We compare G-ML (k-NN classifier with $k = 4$) with three state-of-the-art semi-supervised kernels: non-parametric spectral kernel (Zhu et al., 2005), diffusion kernel (Kondor & Lafferty, 2002), and maximal-alignment kernel (Lanckriet et al., 2004). For all four semi-supervised learning models we use 10-NN unweighted graphs on all the datasets. The non-parametric spectral kernel uses the first 200 eigenvectors (Zhu et al., 2005), whereras G-ML uses the first 20 eigenvectors to form M. For the three competitor semi-supervised kernels, we use support vector machines (one-vs-all classification). We also compare against three standard kernels: RBF kernel (bandwidth learned using 5-fold cross validation), linear kernel, and quadratic kernel. We use the diffusion kernel $K = \exp(-tL)$ with $t = 0.1$ for initializing our alternating minimization algorithm. Note that the various parameter values are set arbitrarily without optimizing and do not give an unfair advantage to the proposed method.

We report the classification error of G-ML averaged over 30 random training/testing splits; the results of competing methods are from (Zhu et al., 2005). The first row of Figure 3 compares error incurred by various methods on each of the four datasets, the second row shows the test error rate at each iteration of G-ML using 30 labeled examples (except for 10 digits dataset where we use 50 examples), while the third row shows the same for 70 labeled examples (100 examples for 10 digits). Clearly, on all the four data sets, G-ML gives comparable or better performance than state-of-the-art semi-supervised learning algorithms and significantly outperforms the supervised learning algorithms.

7.3. Colored Dimensionality Reduction

Next, we apply our method to the task of semi-supervised non-linear dimensionality reduction. We evaluate our method on standard USPS digits dataset, and compare it

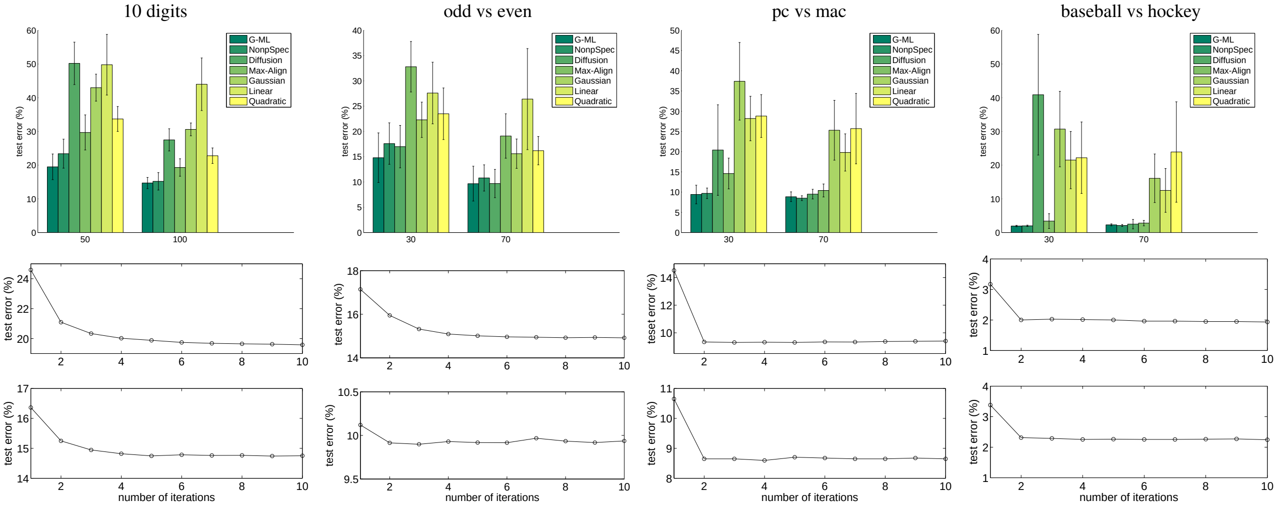

Figure 3. **Top row**: Classification error for various methods on four standard datasets using different number of labeled samples. Note that G-ML consistently performs comparably or better than the best semi-supervised learning methods and significantly outperforms the supervised learning methods. **Middle Row** and **Bottom row**: Classification error rate with 30 labeled samples and 70 labeled data (50, 100 for 10 digits) as the number of iterations increase. In both the cases G-ML improves over the initial (diffusion) kernel .

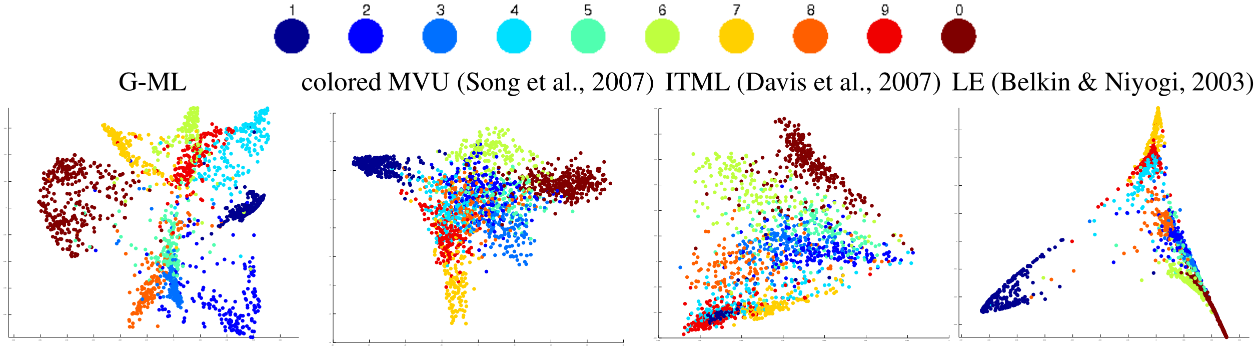

Figure 4. Two dimensional embedding of 2007 USPS digits using different methods. Color of the dots represents different classes of digits (color coding is provided in the top row). We observe that compared to other methods, our method separates the respective manifolds of different digits more accurately, *e.g.* digit 4. (Better viewed in color)

to the state-of-the-art colored Maximum Variance Unfolding (colored MVU) (Song et al., 2007) method which also performs dimensionality reduction for labelled data. We also compare our method to ITML (Davis et al., 2007) that does not take the local geometry into account and Laplacian Eigenmaps (Belkin & Niyogi, 2003) that does not exploit the label information. For visualization, we reduce the dimensionality of the data to two and plot each of the 10 classes of digits with different color (Figure 4). For the proposed G-ML method, we use 200 samples to generate the pairwise constraints, while colored MVU is supplied with all the labels. Note that other than digit 5, G-ML is able to separate manifolds of all the digits in the two-dimensional embedding. In contrast, colored MVU is unable to clearly separate manifolds of digits 4, 5, 8, and 2 while using more labels than the proposed G-ML method.

7.4. Manifold Alignment

In this experiment, we evaluate our method for the task of manifold alignment (Section 6.2) on two datasets, each associated with a different type of transformation. The first dataset consists of images of two subjects sampled from the Yale face B dataset, each with 64 different illumination conditions (varying angles of two illumination sources). Note that the images of each of the subjects lie on an arbitrary oriented two-dimensional manifold. In order to align the two manifolds, we randomly sample 10 must-links for the images with the same illumination conditions. The top row of Figure 5 shows three-dimensional embedding of the images using Laplacian Eigenmaps (Belkin & Niyogi, 2003), proposed G-ML method at various iterations, and ITML method with RBF kernel as the baseline kernel (Davis et al., 2007). We observe that G-ML is able to capture the manifold structure of the Lie group and successfully align them within five iterations. Next, we apply

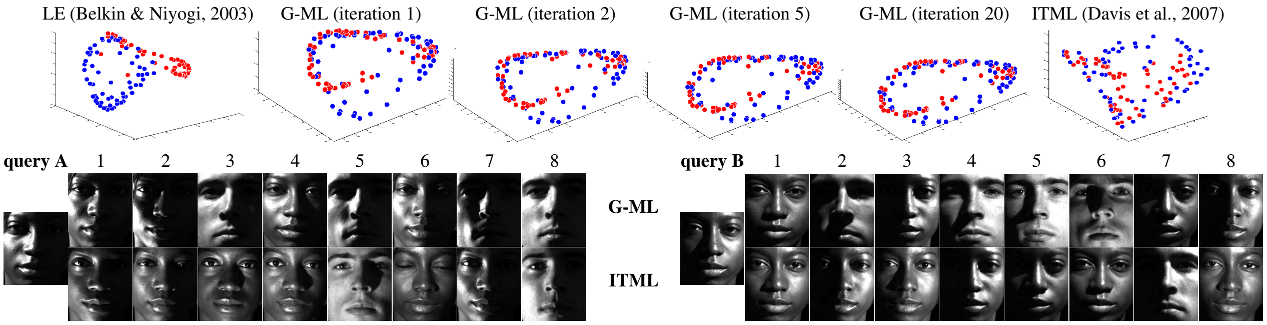

Figure 5. Manifold alignment results (Yale Face). **Top Row:** 3-dimensional embedding of the images of two subjects with different illumination. **Middle Row:** the retrieval result for two queries based on kernel learned using G-ML. **Bottom Row:** the retrieval result for the same two queries using ITML kernel. We observe that G-ML is able to capture the local geometry of the manifold, which is further confirmed by the illumination retrieval results, where unlike ITML, G-ML is able to retrieve similar illumination images irrespective of the subject. (Better viewed in color)

our method to the task of illumination estimation, where the goal is to retrieve the image with the most similar illumination to the given query image. As shown in the middle row of Figure 5, G-ML is able to accurately retrieve similar illumination images irrespective of the identity of the person. The ITML method, which does not capture the local geometry of the unsupervised data, is unable to align the data points w.r.t. the illumination transform and hence unable to accurately retrieve similar illumination images.

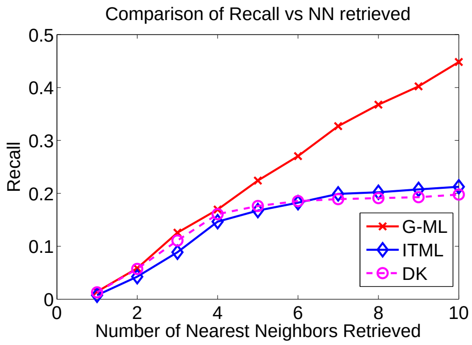

Figure 6. This plot shows recall as a function of number of retrieved images, for G-ML, ITML, and Diffusion Kernel (DK).

To give a quantitative evaluation of manifold alignment, we also performed a similar experiment on a subset of COIL-20 data datasets, which contains images of three subjects with different degree of rotation (72 points uniformly sampled from 0∼360 degree). Images of each subjects should lie on a circular one-dimensional manifold. We apply our method to retrieve images with similar "angle" to a given query image. Figure 6 shows that with 10 randomly chosen similarity-constraints, our method is able to obtain recall of 0.47, significantly outperforming the ITML (0.24) and the diffusion kernel (Kondor & Lafferty, 2002) method (0.23).

Acknowledgements

This research was supported by NSF grants IIS-0713142 and CCF-0728879. ZL is supported by the ICES postdoctoral fellowship from the University of Texas at Austin.

References

Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, *15*, 1373–1396.

Bengio, Y., Delalleau, O., Roux, N. L., Paiement, J., Vincent, P., & Ouimet, M. (2004). Learning eigenfunctions links spectral embedding and kernel PCA. *Neural Computation*, *16*, 2004.

Chapelle, O., Weston, J., & Schlkopf, B. (2003). Cluster kernels for semi-supervised learning. *NIPS*.

Davis, J., Kulis, B., Jain, P., Sra, S., & Dhillon, I. (2007). Information-theoretic metric learning. *ICML*.

Globerson, A., & Roweis, S. (2005). Metric Learning by Collapsing Classes. *NIPS*.

Kondor, R. I., & Lafferty, J. D. (2002). Diffusion kernels on graphs and other discrete input spaces. *ICML* (pp. 315–322).

Lafferty, J., & Lebanon, G. (2005). Diffusion kernels on statistical manifolds. *PAMI*, *6*, 129–163.

Lanckriet, G. R. G., Cristianini, N., Bartlett, P. L., Ghaoui, L. E., & Jordan, M. I. (2004). Learning the kernel matrix with semidefinite programming. *JMLR*, *5*, 27–72.

Ong, C. S., Smola, A. J., & Williamson, R. C. (2005). Learning the kernel with hyperkernels. *JMLR*, *6*, 1043–1071.

Seeger, M. (2002). Covariance kernels from Bayesian generative models. *NIPS*.

Seeger, M. (2008). Cross-validation optimization for large scale structured classification kernel methods. *JMLR*, *9*, 1147–1178.

Sindhwani, V., Niyogi, P., & Belkin, M. (2005). Beyond the point cloud: from transductive to semi-supervised learning. *ICML*.

Song, L., Smola, A., Borgwardt, K. M., & Gretton, A. (2007). Colored maximum variance unfolding. *NIPS*.

Sonnenburg, S., Rätsch, G., Schäfer, C., & Schölkopf, B. (2006). Large scale multiple kernel learning. *JMLR*, *7*, 1531–1565.

Sun, L., Ji, S., & Ye, J. (2008). Adaptive diffusion kernel learning from biological networks for protein function prediction. *Bioinformatics*, *9*.

Weinberger, K., Blitzer, J., & Saul, L. (2006). Distance metric learning for large margin nearest neighbor classification. *NIPS*.

Weinberger, K., Sha, F., & Saul, L. (2004). Learning a kernel matrix for nonlinear dimension reduction. *ICML*.

Xing, E., Ng, A., Jordan, M., & Russell, S. (2002). Distance metric learning, with application to clustering with side-information. *NIPS*.

Zhu, X., Kandola, J., Ghahramani, Z., & Lafferty, J. (2005). Nonparametric transforms of graph kernels for semi-supervised learning. *NIPS*.