Kernel Selection for Semi-Supervised Kernel Machines

Guang Dai DAIGUANG@CSE.UST.HK
Dit-Yan Yeung DYYEUNG@CSE.UST.HK
Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China

Abstract

Existing semi-supervised learning methods are mostly based on either the cluster assumption or the manifold assumption. In this paper, we propose an integrated regularization framework for semi-supervised kernel machines by incorporating both the cluster assumption and the manifold assumption. Moreover, it supports kernel learning in the form of kernel selection. The optimization problem involves joint optimization over all the labeled and unlabeled data points, a convex set of basic kernels, and a discrete space of unknown labels for the unlabeled data. When the manifold assumption is incorporated, graph Laplacian kernels are used as the basic kernels for learning an optimal convex combination of graph Laplacian kernels. Comparison with related methods on the USPS data set shows very promising results.

1. Introduction

In a wide range of real-world machine learning applications, unlabeled data are much easier and cheaper to obtain than labeled data. This characteristic has motivated a surge of research interest over the past decade in the so-called *semi-supervised learning* (SSL) paradigm. By incorporating unlabeled data into the learning process, SSL methods aim at improving the learning performance. This approach is particularly promising in applications where the amount of labeled data available is very limited.

In general, many SSL methods are based on either one or both of two geometric assumptions about the data distribution. The first of these is called the manifold assumption, which assumes the data points as forming a low-dimensional manifold in some input space. The SSL methods based on this assumption typically use the graph Laplacian of a graph-based representation to characterize the manifold structure, e.g., (Zhu et al., 2003; Belkin & Niyogi, 2004; Zhou et al., 2004; Sindhwani et al., 2005; Zhang & Ando, 2005; Zhou et al., 2005; Zhu et al., 2005; Argyriou et al., 2006; Belkin et al., 2006). The second one is called the cluster assumption, which favors decision boundaries for classification passing through low-density regions in the input space (Joachims, 1999; Chapelle & Zien, 2005; Chapelle et al., 2006; Sindhwani et al., 2006).

There is a misconception that SSL is equivalent to transductive learning. While some SSL methods only support transductive inference (Zhu et al., 2003; Belkin & Niyogi, 2004; Zhou et al., 2004; Chapelle & Zien, 2005; Zhang & Ando, 2005; Zhou et al., 2005; Zhu et al., 2005; Argyriou et al., 2006), some others also support inductive inference making out-of-sample extension possible (Joachims, 1999; Sindhwani et al., 2005; Belkin et al., 2006; Chapelle et al., 2006; Sindhwani et al., 2006). In general, SSL methods that support inductive inference are based on kernel methods.

For kernel methods, it is well known that the kernel plays an essential role. A poor kernel choice will lead to impaired performance. Over the past few years, some methods have been proposed for kernel learning or kernel selection. Early methods are limited to learning the parameters of some standard kernel forms (Chapelle et al., 2002). More recent efforts attempt to learn the kernel itself in a nonparametric manner, e.g., by semi-definite programming (SDP) (Lanckriet et al., 2004) or joint minimization over regularization frameworks (Argyriou et al., 2005; Micchelli & Pontil, 2005). In addition, some recent SSL methods (Zhu et al., 2005; Argyriou et al., 2006) seek to learn an optimal kernel based on the graph Laplacian to capture the geometric structure of the

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

data.

In this paper, we propose a new kernel selection method for semi-supervised kernel machines based on the cluster assumption. The proposed method is based on joint minimization over the data points, a convex set of kernels, and a discrete space of unknown labels. Moreover, we also propose a transductive regularization framework that effectively combines the cluster assumption and the manifold assumption. This integrated regularization framework gives a powerful approach to the learning of optimal convex combinations of basic graph Laplacian kernels.

1.1. Notation

For notational consistency, we adopt the following convention throughout this paper.

Let $\mathcal{L} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$ be the labeled data set where $y_i \in \{+1, -1\}$ denotes the class label of $\mathbf{x}_i$, and $\mathcal{U} = \{\mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u}\}$ be the unlabeled data set with $l + u = n$. We assume that all data points $\mathbf{x}_i$ are from the Euclidean space $\mathbb{R}^m$. Under the SSL setting, the goal is to learn a classifier based on both $\mathcal{L}$ and $\mathcal{U}$.

The Euclidean space $\mathbb{R}^m$ is mapped to a high-dimensional (possibly infinite-dimensional) *reproducing kernel Hilbert space* (RKHS) $\mathcal{H}$ via an implicit nonlinear mapping $\phi : \mathbb{R}^m \to \mathcal{H}$. The inner product in $\mathcal{H}$ corresponds to a symmetric kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}}$ which satisfies the finitely positive semidefinite property. In addition, for any $f \in \mathcal{H}$, $f(\mathbf{x}) = \langle f(\cdot), k(\mathbf{x}, \cdot) \rangle_{\mathcal{H}}$, which is called the reproducing property.

2. Semi-Supervised Kernel Machines

We first review the common regularization framework for kernel machines such as support vector machine (SVM) under the standard supervised learning setting. Given a labeled data set $\mathcal{L}$, the goal is to find a real-valued function f by minimizing a regularized loss function that maintains a tradeoff between an empirical loss term and a model-complexity penalty term:

$$\min_{f \in \mathcal{H}_k} \left\{ \sum_{i=1}^{l} L(y_i, f(\mathbf{x}_i)) + \gamma \|f\|_k^2 \right\}, \tag{1}$$

where k is a kernel, $\mathcal{H}_k$ is the RKHS corresponding to k, f is a real-valued function in $\mathcal{H}_k$, $L(\cdot, \cdot)$ is a loss function penalizing the prediction error of f on the labeled data points in $\mathcal{L}$, γ is a regularization parameter, and $\|\cdot\|_k$ denotes the norm in $\mathcal{H}_k$ acting as a regularization term. From the theory of RKHS, we know that the solution f to the optimization problem (1) has the form $f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i k(\mathbf{x}_i, \mathbf{x})$. In other words, the solution can be expressed as an expansion of a subset of data points in $\mathcal{L}$ with nonzero coefficient α_i.

To extend this regularization framework to the SSL setting, we add an extra empirical loss term for the unlabeled data in $\mathcal{U}$. The resulting optimization problem becomes:

$$\min_{f \in \mathcal{H}_k} \left\{ \sum_{i=1}^{l} L(y_i, f(\mathbf{x}_i)) + \beta \sum_{i=l+1}^{l+u} \widetilde{L}(\hat{y}_i, f(\mathbf{x}_i)) + \gamma \|f\|_k^2 \right\}$$
$$\text{subject to: } \frac{1}{u} \sum_{i=l+1}^{l+u} \max(0, \hat{y}_i) = r, \tag{2}$$

where $\widetilde{L}(\cdot, \cdot)$ is the additional loss function, $\hat{y}_i$ is the predicted label for unlabeled data point $\mathbf{x}_i \in \mathcal{U}$, β is a positive parameter for the new loss function, and r is the class ratio representing the percentage of positive examples in $\mathcal{U}$ assumed to be known in advance. One possible way of defining the loss function $\widetilde{L}$ is $\widetilde{L}(\hat{y}_i, f(\mathbf{x}_i)) = \min\{L(+1, f(\mathbf{x}_i)), L(-1, f(\mathbf{x}_i))\}$, meaning that $\hat{y}$ is set in such a way that $L(\hat{y}_i, f(\mathbf{x}_i))$ is minimized. Note that an unlabeled data point $\mathbf{x}_i$ is penalized more if $f(\mathbf{x}_i)$ is closer to zero. The solution to this optimization problem essentially implements the cluster assumption so that decision boundaries passing through low-density regions are preferred. The SSL models based on this idea include (Joachims, 1999; Chapelle & Zien, 2005; Chapelle et al., 2006; Sindhwani et al., 2006).

In what follows, we will refer to kernel machines based on the problem (2) as *semi-supervised kernel machines*. Even though the loss function L is typically chosen to be a convex function, $\widetilde{L}$ is non-convex and optimization has to be performed over the discrete variables $\hat{y}_i$. The optimization problem as a whole is thus non-convex.

3. Kernel Selection for Semi-Supervised Kernel Machines

As discussed in Section 1, introducing adaptability or learning ability into kernel design can improve the performance of kernel methods. In this section, we propose an extension to the optimization problem in (2) for semi-supervised kernel machines by incorporating a specific type of kernel learning called kernel selection.

3.1. Optimization Problem

A promising recent approach to kernel learning is to learn an optimal convex combination of some prespecified basic kernels. We apply this approach here to extend the optimization problem in (2) for semi-supervised kernel machines. The corresponding opti-

mization problem can be stated as follows:

$$\min_{k\in\mathcal{K}}\min_{f\in\mathcal{H}_k}\left\{\sum_{i=1}^{l}L(y_i,f(\mathbf{x}_i))+\beta\sum_{i=l+1}^{l+u}\widetilde{L}(\hat{y}_i,f(\mathbf{x}_i))+\gamma\|f\|_k^2\right\}$$
$$\text{subject to:}\quad \frac{1}{u}\sum_{i=l+1}^{l+u}\max(0,\hat{y}_i)=r, \tag{3}$$

where $\mathcal{K}$ is a convex set of kernels. Specifically, we consider one possibility where each element in $\mathcal{K}$ is a convex combination of N basic kernels, i.e., $\mathcal{K}=\left\{\sum_{i=1}^{N}\lambda_i k_i:\sum_{i=1}^{N}\lambda_i=1,\lambda_i\geq 0,i=1,\ldots,N\right\}$. Note that this is a complicated joint optimization problem over k, f and $\{\hat{y}_i\}_{i=l+1}^{l+u}$. For simplicity, we denote $\{\hat{y}_i\}_{i=l+1}^{l+u}$ by $\hat{\mathbf{y}}_{\mathcal{U}}$.

3.2. A Deterministic Annealing Approach

The non-convex optimization problem in (3) involving discrete variables $\hat{\mathbf{y}}_{\mathcal{U}}$ and non-convex loss function $\widetilde{L}$ is difficult to solve directly. We use a *deterministic annealing* (DA) approach to solve this problem. In essence, DA is a homotopy approach for dealing with combinatorial optimization problems. Instead of solving the original problem directly, a DA method relaxes the original problem to a simpler but related problem. The simplified problem is typically a convex problem that is much easier to solve and has guarantee for global optimality. The simplified problem is then gradually deformed to the original problem by varying a temperature parameter T. Recently, Sindhwani et al. (2006) proposed a DA method for semi-supervised kernel machines.

We apply DA to gradually approach the global solution and simultaneously learn an optimal convex combination of the basic kernels. In DA, discrete variables are usually formulated as random variables over a space of probability distributions. The original optimization problem is relaxed into minimizing the expectation of the original objective function with respect to a probability distribution over discrete variables. Inspired by (Sindhwani et al., 2006), we express the relaxed optimization problem as:

$$\min_{k\in\mathcal{K}}\min_{f\in\mathcal{H}_k}\min_{\mathbf{p}\in\mathcal{P}(\{+1,-1\}^u)}\{E_{\mathbf{p}}\mathcal{C}_{\beta,\gamma}(k,f,\hat{\mathbf{y}}_{\mathcal{U}})-T\,s(\mathbf{p})\}$$
$$\text{subject to:}\quad \frac{1}{u}\sum_{i=1}^{u}p_i=r, \tag{4}$$

where $\mathcal{C}_{\beta,\gamma}(k,f,\hat{\mathbf{y}}_{\mathcal{U}})$ is a short form for the objective function in (3), $\mathcal{P}$ is a family of probability distributions over the discrete variables $\hat{\mathbf{y}}_{\mathcal{U}}$, T is a temperature parameter, $\mathbf{p}=(p_1,\ldots,p_u)$ with each p_i representing the probability that $\hat{y}_{l+i}=1$, $E_{\mathbf{p}}$ is the expectation with respect to the probability distribution $\mathbf{p}$, and $s(\mathbf{p})$ is the entropy over $\mathbf{p}$. By computing the expectation in (4), the relaxed optimization problem for DA is rewritten as:

$$\min_{k\in\mathcal{K}}\min_{f\in\mathcal{H}_k}\min_{\mathbf{p}\in\mathcal{P}(\{+1,-1\}^u)}\left\{\sum_{i=1}^{l}L(y_i,f(\mathbf{x}_i))+\right.$$
$$\beta\sum_{i=1}^{u}p_iL(+1,f(\mathbf{x}_{l+i}))+\beta\sum_{i=1}^{u}(1-p_i)L(-1,f(\mathbf{x}_{l+i}))+$$
$$\left.\gamma\|f\|_k^2+T\sum_{i=1}^{u}p_i\log p_i+T\sum_{i=1}^{u}(1-p_i)\log(1-p_i)\right\}$$
$$\text{subject to:}\quad \frac{1}{u}\sum_{i=1}^{u}p_i=r. \tag{5}$$

We notice from (5) that for a fixed kernel k (hence RKHS $\mathcal{H}_k$), the globally optimal solution for f can be effectively tracked by DA as T tends to zero. For fixed distribution $\mathbf{p}$, (5) can be reduced to a joint optimization problem over both data in $\mathcal{L}$ and $\mathcal{U}$ and the convex set of kernels $\mathcal{K}$, and the corresponding solution can be found based on the recently developed kernel selection framework for the supervised learning setting (Argyriou et al., 2005; Micchelli & Pontil, 2005). For any given k (hence $\mathcal{H}_k$) and function f, problem (5) becomes a convex problem over the variables $\mathbf{p}$. Similar to (Sindhwani et al., 2006), the above observations motivate us to optimize problem (5) via a joint minimization over both the data and the convex set of kernels for fixed $\mathbf{p}$ and over $\mathbf{p}$ for fixed k and f. We consider these two subproblems in detail below.

Optimization on fixed $\mathbf{p}$

For fixed $\mathbf{p}$, problem (5) degenerates to:

$$\min_{k\in\mathcal{K}}\min_{f\in\mathcal{H}_k}\left\{\sum_{i=1}^{l}L(y_i,f(\mathbf{x}_i))+\beta\sum_{i=1}^{u}p_iL(+1,f(\mathbf{x}_{l+i}))\right.$$
$$\left.+\beta\sum_{i=1}^{u}(1-p_i)L(-1,f(\mathbf{x}_{l+i}))+\gamma\|f\|_k^2\right\}. \tag{6}$$

Based on all the training data in $\mathcal{L}$ and $\mathcal{U}$, we define the RKHS $\mathcal{H}_k$ for a given kernel k as the completion of the span of the functions $k(\mathbf{x}_i,\cdot)$ for all data points in $\mathcal{L}$ and $\mathcal{U}$, i.e., $\mathcal{H}_k=\overline{span\{k(\mathbf{x}_1,\cdot),\ldots,k(\mathbf{x}_n,\cdot)\}}$. In general $\mathcal{H}_k$ should depend on k only, not on the data points. So this may be seen as an "empirical" version of the RKHS. Furthermore, for any fixed kernel k (hence $\mathcal{H}_k$), it follows from the representer theorem that if f is a solution to the problem (6), then it has the form $f(\mathbf{x})=\sum_{i=1}^{n}\alpha_i k(\mathbf{x}_i,\mathbf{x})$. Based on this form, the squared norm $\|f\|_k^2$ can be computed as $\|f\|_k^2=\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha}$ where $\boldsymbol{\alpha}=(\alpha_1,\ldots,\alpha_n)^T$ and $\mathbf{K}$ is an $n\times n$ Gram matrix induced by a kernel $k\in\mathcal{K}$ on $\mathcal{L}$ and $\mathcal{U}$. As a result, the optimization problem (6) can

be converted into the following form:

$$\min_{k\in\mathcal{K}}\min_{\boldsymbol{\alpha}\in\mathbb{R}^n}\left\{\sum_{i=1}^{l}L(y_i,\boldsymbol{\alpha}^T\mathbf{k}_i)+\beta\sum_{i=1}^{u}p_iL(+1,\boldsymbol{\alpha}^T\mathbf{k}_{l+i})\right.$$
$$\left.+\beta\sum_{i=1}^{u}(1-p_i)L(-1,\boldsymbol{\alpha}^T\mathbf{k}_{l+i})+\gamma\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha}\right\}, \quad (7)$$

where $\mathbf{k}_i=(k(\mathbf{x}_1,\mathbf{x}_i),\ldots,k(\mathbf{x}_n,\mathbf{x}_i))^T$ is the ith column of $\mathbf{K}$. Since each p_i fixed here belongs to the range $[0,1]$, from the recent results on learning convex combinations of kernels via regularization (Argyriou et al., 2005; Micchelli & Pontil, 2005), we know that the optimal kernel for the problem (7) can always be computed as a convex combination of at most $n+2$ basic kernels, even though $\mathcal{K}$ is an uncountable set. Through the problem (7), we can effectively extend the theoretical study of (Argyriou et al., 2005; Micchelli & Pontil, 2005) on kernel selection to the SSL setting. We now rewrite the optimization problem (7) as an equivalent saddle point problem:

$$\max_{k\in\mathcal{K}}\min_{\boldsymbol{\alpha}\in\mathbb{R}^n}V(\boldsymbol{\alpha},\mathbf{K}), \quad (8)$$

where the new function $V(\boldsymbol{\alpha},\mathbf{K})$ is defined as

$$V(\boldsymbol{\alpha},\mathbf{K})=\sum_{i=1}^{l}L^*(y_i,\alpha_i)+\sum_{i=1}^{u}\widetilde{L}^*(\pm1,\alpha_{l+i})+\frac{1}{4\gamma}\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha}.$$

Here, L^* and $\widetilde{L}^*$ denote the corresponding conjugate functions of the loss functions for the labeled and unlabeled data sets, respectively. Following (Rockafellar, 1996), both can be further defined as $L^*(y_i,\alpha)=\sup\{\theta\alpha-L(y_i,\theta):\alpha,\theta\in\mathbb{R}\}$ and $\widetilde{L}^*(\pm1,\alpha)=\sup\{\theta\alpha-\beta p_iL(+1,\theta)-\beta(1-p_i)L(-1,\theta):\alpha,\theta\in\mathbb{R}\}$.

Based on similar analysis as in (Argyriou et al., 2005), we can obtain the necessary and sufficient conditions for a pair $(\boldsymbol{\alpha},\mathbf{K})$ to be a saddle point, motivating us to adopt a greedy algorithm to solve the problem. The greedy algorithm begins with an initial kernel matrix $\mathbf{K}_{(1)}$ and then solves the optimization problem $\min_{\boldsymbol{\alpha}\in\mathbb{R}^n}V(\boldsymbol{\alpha},\mathbf{K}_{(1)})$ for the corresponding vector $\boldsymbol{\alpha}_{(1)}$. In general, based on the ith kernel matrix $\mathbf{K}_{(i)}$ and the vector $\boldsymbol{\alpha}_{(i)}$ obtained from solving the corresponding optimization problem, we seek a new basic kernel $\mathbf{K}_{(o)}$ satisfying $\boldsymbol{\alpha}_{(i)}^T\mathbf{K}_{(o)}\boldsymbol{\alpha}_{(i)}>\boldsymbol{\alpha}_{(i)}^T\mathbf{K}_{(i)}\boldsymbol{\alpha}_{(i)}$. Then the $(i+1)$th kernel $\mathbf{K}_{(i+1)}$ is calculated as $\mathbf{K}_{(i+1)}=\hat{\mu}\mathbf{K}_{(o)}+(1-\hat{\mu})\mathbf{K}_{(i)}$, which is a convex combination of $\mathbf{K}_{(o)}$ and $\mathbf{K}_{(i)}$. The coefficient $\hat{\mu}$ for the convex combination is calculated as $\hat{\mu}=\arg\max_{\mu\in(0,1]}V(\boldsymbol{\alpha}_{(i)},\mu\mathbf{K}_{(o)}+(1-\mu)\mathbf{K}_{(i)})$. By repeating this iterative procedure, we can obtain a sequence of kernels $\langle\mathbf{K}_{(1)},\ldots,\mathbf{K}_{(t)}\rangle$. Some remarks are given here. If the set $\mathcal{K}$ is a convex combination of a finite number of basic kernels, then the corresponding combination coefficients $\lambda_1,\ldots,\lambda_N$ can be identified by tracking $\hat{\mu}$ for each kernel matrix, and $\mathbf{K}_{(o)}$ can be calculated on a basic kernel $k_c\in\mathcal{K}$. If $\mathcal{K}$ is a convex combination of continuously parameterized basic kernels, then we just track the change of the corresponding parameter to calculate $\mathbf{K}_{(o)}$. More detailed discussions can be found in (Argyriou et al., 2005). In our experiments, since L is a squared loss function, $\boldsymbol{\alpha}_{(i)}$ has a closed-form solution and $\hat{\mu}$ is computed by the Newton method.

Optimization on fixed k and f

After obtaining an optimal saddle point $(\hat{\boldsymbol{\alpha}},\hat{\mathbf{K}})$ for the optimization problem (8) based on a fixed $\mathbf{p}$, the optimization problem (5) is reformulated as:

$$\min_{\mathbf{p}\in\mathcal{P}(\{+1,-1\}^u)}J(\mathbf{p})$$
$$\text{subject to: }\ \frac{1}{u}\sum_{j=1}^{u}p_i=r, \quad (9)$$

where the function $J(\mathbf{p})$ is expressed as

$$J(\mathbf{p})=$$
$$\beta\sum_{i=1}^{u}p_iL(+1,\hat{\boldsymbol{\alpha}}^T\hat{\mathbf{k}}_{l+i})+\beta\sum_{i=1}^{u}(1-p_i)L(-1,\hat{\boldsymbol{\alpha}}^T\hat{\mathbf{k}}_{l+i})$$
$$+T\sum_{i=1}^{u}p_i\log p_i+T\sum_{i=1}^{u}(1-p_i)\log(1-p_i). \quad (10)$$

Here, $\hat{\mathbf{k}}_i$ denotes the ith column of $\hat{\mathbf{K}}$. Since (9) is convex with respect to $\mathbf{p}$, its solution can be found easily.

For a fixed temperature parameter T, the optimization problem (4) effectively selects an optimal combination of kernels and also relaxes the non-convex combinatorial optimization problem into another problem that is easier to solve. We decrease the temperature parameter T gradually according to the DA procedure until some termination conditions are satisfied. Algorithm 1 summarizes the major steps of the kernel selection algorithm for semi-supervised kernel machines.

3.3. Combining Graph Laplacian Kernels for Semi-Supervised Kernel Machines

In this subsection, we further extend the above framework formulated based on the cluster assumption to incorporate the manifold assumption as well.

We first construct a neighborhood graph $G=(\mathcal{V},\mathcal{E})$ to represent the local geometric structure of the data based on pairwise relationships. $\mathcal{V}$ is the vertex set for

Algorithm 1 Learning an Optimal Convex Combination of Basic Kernels for Semi-Supervised Kernel Machines

Input: $\mathcal{L} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$, $\mathcal{U} = \{\mathbf{x}_i\}_{i=1+l}^{n}$, β, γ, r, r_0, T_0, ϵ_0, T_{max};
Initialize: $\mathbf{p} = (r, \ldots, r) \in \mathbb{R}^u$, $\mathbf{p}' = \mathbf{p}$, $T = T_0$;
Select $\hat{\mathbf{K}}$; % induced by any kernel $k \in \mathcal{K}$ on $\mathcal{L}$ and $\mathcal{U}$
repeat
  repeat
    $\mathbf{K}_{(1)} \leftarrow \hat{\mathbf{K}}$;
    for $i = 1, \ldots, T_{max}$ **do**
      $\boldsymbol{\alpha}_{(i)} = \arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} V(\boldsymbol{\alpha}, \mathbf{K}_{(i)})$;
      Find $\mathbf{K}_{(o)}$ s.t. $\boldsymbol{\alpha}_{(i)}^T \mathbf{K}_{(o)} \boldsymbol{\alpha}_{(i)} > \boldsymbol{\alpha}_{(i)}^T \mathbf{K}_{(i)} \boldsymbol{\alpha}_{(i)}$;
      if such $\mathbf{K}_{(o)}$ does not exist **then**
        break;
      end if
      $\hat{\mu} = \arg\max_{\mu \in (0,1]} V(\boldsymbol{\alpha}_{(i)}, \mu \mathbf{K}_{(o)} + (1-\mu)\mathbf{K}_{(i)})$;
      $\mathbf{K}_{(i+1)} \leftarrow \hat{\mu} \mathbf{K}_{(o)} + (1 - \hat{\mu}) \mathbf{K}_{(i)}$;
    end for
    $\hat{\boldsymbol{\alpha}} \leftarrow \boldsymbol{\alpha}_{(i)}$;
    $\hat{\mathbf{K}} \leftarrow \mathbf{K}_{(i)}$;
    $\mathbf{p}' \leftarrow \mathbf{p}$;
    Calculate $\mathbf{p}$ using $\hat{\boldsymbol{\alpha}}$ and $\hat{\mathbf{K}}$;
  until $||p' - p|| < \epsilon_0$
  $T \leftarrow T / r_0$;
until $t < \epsilon_0$
Output: $\hat{\mathbf{K}}$, $\hat{\boldsymbol{\alpha}}$, and $f(\mathbf{x}) = \sum_{i=1}^{n} \hat{\alpha}_i \hat{k}(\mathbf{x}_i, \mathbf{x})$.

all data points $\mathbf{x}_i$ in $\mathcal{L}$ and $\mathcal{U}$, and $\mathcal{E}$ is the edge set where each edge e_{ij} has a weight w_{ij} representing the pairwise relationship between vertices $\mathbf{x}_i$ and $\mathbf{x}_j$. The corresponding affinity matrix $\mathbf{W} = [w_{ij}]_{n \times n}$ is defined based on the heat kernel as:

$$w_{ij} = \begin{cases} \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / \sigma^2), & \text{if } \mathbf{x}_j \in \mathcal{N}_K(\mathbf{x}_i) \\ & \text{or } \mathbf{x}_i \in \mathcal{N}_K(\mathbf{x}_j); \\ 0, & \text{otherwise,} \end{cases} \tag{11}$$

where $\mathcal{N}_K(\mathbf{x})$ denotes the set of K nearest neighbors of $\mathbf{x}$, σ is a positive constant, and $\|\mathbf{x}\|$ denotes the norm of $\mathbf{x}$. The graph Laplacian matrix $\mathbf{L}$ is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is a diagonal matrix with the diagonal entries $d_{ii} = \sum_j w_{ij}$. $\mathbf{L}$ can be regarded as an empirical version of the Laplace-Beltrami operator if the underlying distribution of the whole data space is a Riemannian manifold. Instead of using $\mathbf{L}$, there also exist other possible choices (Belkin et al., 2006), such as iterated graph Laplacian $\mathbf{L}^p$ ($p > 0$), heat semigroup $e^{-t\mathbf{L}^p}$, normalized graph Laplacian $\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$, etc. In what follows, the graph Laplacian $\mathbf{L}$ may be chosen to be any of these forms.

For the defined graph G, we consider a real-valued vector $\mathbf{f} = (f_1, \ldots, f_n) \in \mathbb{R}^n$ as representing the label information for the data points in G. In general, we only consider $\mathbf{f}$ in the space $\mathcal{H}_G$ that is orthogonal to the eigenvectors of $\mathbf{L}$ with zero eigenvalues. The corresponding graph regularization term can be induced by the strict norm for $\mathbf{f}$ under its inner product: $\|\mathbf{f}\|_{\mathcal{H}_G}^2 = \langle \mathbf{f}, \mathbf{f} \rangle_{\mathcal{H}_G} = \mathbf{f}^T \mathbf{L} \mathbf{f}$. Moreover, from the definition of graph Laplacian matrix above, we can also express it as $\|\mathbf{f}\|_{\mathcal{H}_G}^2 = \frac{1}{2} \sum_{i,j=1}^{n} (f_i - f_j)^2 w_{ij}$. By integrating this into the regularization framework presented above for semi-supervised kernel machines, the following optimization problem can be formulated for finding a more effective $\mathbf{f}$ that also captures the manifold assumption:

$$\min_{\mathbf{f} \in \mathcal{H}_G} \left\{ \sum_{i=1}^{l} L(y_i, f_i) + \beta \sum_{i=l+1}^{l+u} \widetilde{L}(\hat{y}_i, f_i) + \gamma \|\mathbf{f}\|_{\mathcal{H}_G}^2 \right\}$$
$$\text{subject to: } \frac{1}{u} \sum_{i=l+1}^{l+u} \max(0, \hat{y}_i) = r. \tag{12}$$

Here, $\hat{y}_i$ may be seen as "hard" labels for unlabeled data points while f_i are "soft" labels for all data points. While existing semi-supervised kernel machines (Joachims, 1999; Chapelle & Zien, 2005; Chapelle et al., 2006; Sindhwani et al., 2006) are based on the cluster assumption and graph-based SSL methods (Zhu et al., 2003; Zhou et al., 2004; Zhou et al., 2005; Belkin et al., 2006) are based on the manifold assumption, the optimization problem in (12) attempts to capture both assumptions into an integrated regularization framework.[1]

According to (Argyriou et al., 2006), the graph Laplacian kernel matrix for $\mathcal{H}_G$ can be induced by the pseudoinverse $\mathbf{L}^+ = [L_{ij}^+]_{i,j=1,\ldots,n}$ of the graph Laplacian matrix $\mathbf{L}$ and $\mathcal{H}_G$ essentially becomes an RKHS. For data points $\mathbf{x}_i$ and $\mathbf{x}_j$, the corresponding graph kernel can be calculated by $k(\mathbf{x}_i, \mathbf{x}_j) = L_{ij}^+$. Furthermore, based on the representer theorem for $\mathcal{H}_G$, we have $f_i = \sum_{j=1}^{n} \alpha_j L_{ij}^+$. Thus, the problem (12) can be rewritten as:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \left\{ \sum_{i=1}^{l} L(y_i, \sum_{j=1}^{n} \alpha_j L_{ij}^+) + \right.$$
$$\left. \beta \sum_{i=l+1}^{l+u} \widetilde{L}(\hat{y}_i, \sum_{j=1}^{n} \alpha_j L_{ij}^+) + \gamma \boldsymbol{\alpha}^T \mathbf{L}^+ \boldsymbol{\alpha} \right\}$$
$$\text{subject to: } \frac{1}{u} \sum_{i=l+1}^{l+u} \max(0, \hat{y}_i) = r. \tag{13}$$

Zhang and Ando (2005) proposed a graph-based SSL

[1] Note that the problem (12) differs from the low-density SSL method of (Chapelle & Zien, 2005) which combines semi-supervised kernel machines with graph-based techniques. In (Chapelle & Zien, 2005), the kernel captures some low-density distance measure defined on the nearest-neighbor graph, so that semi-supervised kernel machines based on this kernel still follow the cluster assumption.

method based on spectral decomposition and graph Laplacian kernels. Its optimization problem for kernel learning can be stated as

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^n}\left\{\sum_{i=1}^{l} L(y_i,\sum_{j=1}^{n}\alpha_j L_{ij}^{+})+\gamma\boldsymbol{\alpha}^T\mathbf{L}^{+}\boldsymbol{\alpha}\right\}, \qquad (14)$$

which is a special case of the optimization problem in (13) above.

Belkin et al. (2006) recently proposed the manifold regularization framework for SSL in which an additional graph regularization term is introduced to capture the manifold structure of the data distribution. Based on graph Laplacian kernels, the conventional norm and the manifold regularization term in (Belkin et al., 2006) can be converted into $\boldsymbol{\alpha}^T\mathbf{L}^{+}\boldsymbol{\alpha}$, and hence the manifold regularization framework can be reformulated in the form (14). Argyriou et al. (2006) employed the graph Laplacian kernel to extend the supervised learning problem in (1) to the SSL setting. Zhang and Ando (2005) showed that the objective function in (Argyriou et al., 2006) can also be formulated in the form (14). As such, the methods in (Argyriou et al., 2006; Belkin et al., 2006) can be seen as special cases of (13) since this more general form is based on both the cluster assumption and the manifold assumption.

Another problem to address is how to specify the initial basic graph Laplacian kernels. Currently there exist two major approaches to this problem. One approach is based on the empirical kernel alignment score with order constraints on the linear combination coefficients (Zhu et al., 2005). Another approach is based on a regularized risk function (essentially the same as that in (14)) on which joint minimization over both the training data and the set of graph Laplacian kernels (Argyriou et al., 2006) is applied. Here, we present a new graph Laplacian kernel selection method based on the problem (13). Considering the overfitting problem encountered by empirical kernel alignment in (Zhu et al., 2005) and the lack of the cluster assumption in (Argyriou et al., 2006), our method is expected to be more general and robust.

Let $\mathbf{L}_1^{+},\ldots,\mathbf{L}_N^{+}$ denote the N graph Laplacian kernels corresponding to N graphs constructed from all the data points. The corresponding convex set is $coL=\{\widehat{\mathbf{L}}^{+}=\sum_{i=1}^{N}\lambda_i\mathbf{L}_i^{+}:\sum_{i=1}^{N}\lambda_i=1,\lambda_i\geq 0,i=1,\ldots,N\}$. Motivated by the kernel selection method for semi-supervised kernel machines presented in Section 3.2, the optimization problem for learning an optimal combination of graph Laplacian kernels can be stated as:

$$\min_{\widehat{\mathbf{L}}\in coL}\min_{\boldsymbol{\alpha}\in\mathbb{R}^n}\left\{\sum_{i=1}^{l} L(y_i,\sum_{j=1}^{n}\alpha_j\widehat{L}_{ij}^{+})+\beta\sum_{i=l+1}^{l+u}\widetilde{L}(\hat{y}_i,\sum_{j=1}^{n}\alpha_j\widehat{L}_{ij}^{+})+\gamma\boldsymbol{\alpha}^T\widehat{\mathbf{L}}^{+}\boldsymbol{\alpha}\right\}$$
$$\text{subject to: } \frac{1}{u}\sum_{i=l+1}^{l+u}\max(0,\hat{y}_i)=r, \qquad (15)$$

where $\widehat{\mathbf{L}}^{+}=[\widehat{L}_{ij}^{+}]_{i,j=1,\ldots,n}$. Obviously, the problem (15) can also be solved efficiently using Algorithm 1 developed for kernel selection of semi-supervised kernel machines.

4. Experimental Evaluation

To evaluate the performance of our proposed methods, we report here some classification experiments on the USPS data set that was also used in (Argyriou et al., 2006) for graph Laplacian kernel selection. For simplicity, we only consider the binary classification problems to compare our methods with some closely related kernel selection methods (Argyriou et al., 2005; Argyriou et al., 2006).

The kernel selection method in (Argyriou et al., 2005) is based on the standard regularization framework (1) and hence it is related to our kernel selection method for semi-supervised kernel machines. On the other hand, the graph Laplacian kernel selection method for SSL in (Argyriou et al., 2006) is related to our graph Laplacian kernel selection method. For the convenience of referencing, KS-SL (kernel selection for supervised learning) refers to the method in (Argyriou et al., 2005) and GLKS-SSL (graph Laplacian kernel selection for semi-supervised learning) refers to the method in (Argyriou et al., 2006). In contrast, our two methods are referred to as KS-SSKM and GLKS-SSKM.

As in (Argyriou et al., 2006), we have included in our experiments five digit pairs $\{(1,7),(2,3),(2,7),(3,8),(4,7)\}$ from the USPS data set. The dimensionality of each digit image and the number of digits in each digit class of each set are 256 and 200, respectively. We partition each data set randomly into disjoint labeled and unlabeled sets. In each labeled set, six examples from each class are labeled. Each experiment is repeated 10 times and the average classification error rate tested on the unlabeled data set is reported. Similar to (Sindhwani et al., 2006), the methods are compared on the effectiveness of optimization under the regularization

parameter values of $10^{-h}\,(h = 1,\ldots,5)$. For our methods, the input parameters are set as follows: $r = 0.5$, $r_0 = 1.4$, $T_0 = 10$, $\epsilon_0 = 10^{-7}$, and $T_{max} = 50$. For each experiment, we try to select a very poor initial kernel for our methods to demonstrate the effectiveness of kernel selection.

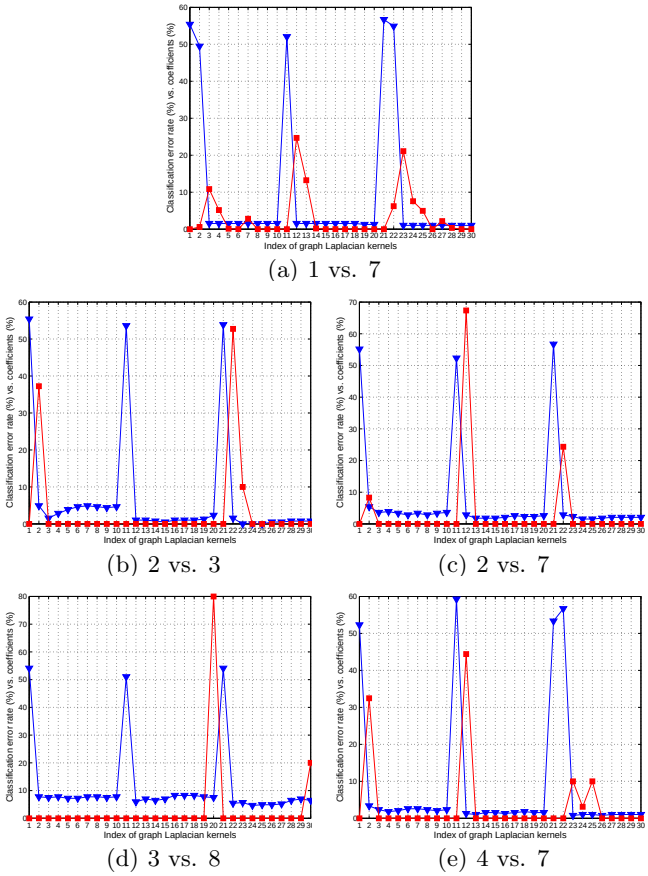
First, we compare our KS-SSKM method with KS-SL (Argyriou et al., 2005). We consider a convex set formed by 20 Gaussian kernels as basic kernels, with their kernel parameters σ equally spaced in the range $[0.1, 100000]$ under log scale. Usually some of these basic kernels are very poor choices and they are included to act as noise to evaluate the robustness of the kernel selection methods. Table 1 shows the results. As expected, KS-SSKM gives significantly lower classification errors than KS-SL since it captures the cluster assumption in its formulation.

Next, we perform experiments to compare our GLKS-SSKM method with GLKS-SSL (Argyriou et al., 2006). We adopt the same setup in (Argyriou et al., 2006) for our experiments. The 30 basic graph Laplacian matrices are created from corresponding graphs constructed based on the k-nearest-neighbor criterion for $k = 1,\ldots,10$ with the Euclidean, affine transformation, and tangent distances. Moreover, following (Argyriou et al., 2006), we run GLKS-SSKM and GLKS-SSL on four different convex sets of graph Laplacian kernels, with three sets based on the three distance metrics and the last one based on all collected metrics. Table 2 summarizes the experimental results. GLKS-SSKM outperforms GLKS-SSL in different cases, showing that integrating the cluster assumption and the manifold assumption does help. In addition, the results of our GLKS-SSKM method on the convex set of all graph Laplacian kernels are very close to (or even better than) the best results of GLKS-SSKM on the other three convex sets, implying that GLKS-SSKM is effective in learning an optimal combination of graph Laplacian kernels.

To demonstrate the kernel selection behavior of our methods in more detail, we also study semi-supervised kernel machines based on each individual basic graph Laplacian kernel. Figure 1 shows the results, where we apply GLKS-SSKM on the convex set of 30 basic graph Laplacian kernels as described above and begin with the "poor" graph kernel denoted by "1". In each subfigure, $(1,\ldots,10)$ in the horizontal axis denote the graph Laplacian kernels based on the k-nearest-neighbor criterion for $k = 1,\ldots,10$ with the Euclidean distance, $(11,\ldots,20)$ denote those with the affine transformation distance, and $(21,\ldots,30)$ for those with the tangent distance. For each graph Laplacian kernel, the classification error rate based on that kernel alone and its combination coefficient for the kernel selection method are shown separately. The high correlation between the two curves shows that the kernel selection procedure focuses on the best kernels effectively. More specifically, while some graph Laplacian kernels such as those denoted by "1", "11" and "21" have very high classification error rates, the corresponding coefficients learned by GLKS-SSKM become very small.

Table 1. Classification error rates (%) and standard deviations when KS-SSKM is compared with KS-SM.

Data sets	KS-SM	KS-SSKM
USPS (1 vs. 7)	01.60±00.72	00.00±00.00
USPS (2 vs. 3)	10.10±04.20	05.10±01.27
USPS (2 vs. 7)	09.28±06.41	01.86±00.43
USPS (3 vs. 8)	10.82±04.09	07.42±00.83
USPS (4 vs. 7)	07.53±03.88	03.92±02.01



(a) 1 vs. 7 (b) 2 vs. 3 (c) 2 vs. 7 (d) 3 vs. 8 (e) 4 vs. 7

Figure 1. Combination coefficients (%) vs. classification error rates (%) on different basic graph Laplacian kernels, where ▼ denotes the classification error rates and ■ denotes the combination coefficients.

Table 2. Classification error rates (%) and standard deviations when GLKS-SSKM is compared with GLKS-SSL.

Data set	Euclidean		Transformation		Tangent		All	
	GLKS-SSL	GLKS-SSKM	GLKS-SSL	GLKS-SSKM	GLKS-SSL	GLKS-SSKM	GLKS-SSL	GLKS-SSKM
USPS (1 vs. 7)	01.55±00.00	01.55±00.00	01.55±00.00	00.77±00.00	01.03±00.00	00.98±00.12	01.19±00.22	00.90±00.14
USPS (2 vs. 3)	03.20±00.76	01.24±00.46	02.68±01.12	00.67±00.35	01.13±00.14	00.15±00.23	02.22±01.10	00.15±00.23
USPS (2 vs. 7)	03.92±00.28	03.51±00.14	02.73±00.50	02.01±00.34	02.11±00.34	01.86±00.22	02.73±00.76	02.01±00.22
USPS (3 vs. 8)	07.06±00.90	06.39±00.84	06.96±00.71	06.13±00.69	06.19±00.58	04.74±00.14	06.39±00.72	04.43±00.34
USPS (4 vs. 7)	02.63±00.56	01.80±00.18	01.91±00.43	01.03±00.32	01.16±00.25	00.93±00.18	02.37±01.28	00.88±00.23

5. Conclusion

We have proposed a novel regularization framework for semi-supervised kernel machines by integrating both the cluster assumption and the manifold assumption. Experimental comparison with related methods proposed recently in the machine learning community demonstrates the effectiveness of this framework in exploiting the geometric structure of the data.

Acknowledgments

This research is supported by Competitive Earmarked Research Grant (CERG) 621706 from the Research Grants Council (RGC) of the Hong Kong Special Administrative Region, China.

References

Argyriou, A., Herbster, M., & Pontil, M. (2006). Combining graph Laplacians for semi–supervised learning. *Advances in Neural Information Processing Systems 18*. MIT Press, Cambridge, MA, USA.

Argyriou, A., Micchelli, C. A., & Pontil, M. (2005). Learning convex combinations of continuously parameterized basic kernels. *Proceedings of the Eighteenth Annual Conference on Learning Theory* (pp. 338–352).

Belkin, M., & Niyogi, P. (2004). Semi-supervised learning on Riemannian manifolds. *Machine Learning*, *56*, 209–239.

Belkin, M., Niyogi, P., & Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, *7*, 2399–2434.

Chapelle, O., Chi, M., & Zien, A. (2006). A continuation method for semi-supervised SVMs. *Proceedings of the Twenty-Third International Conference on Machine Learning*.

Chapelle, O., Vapnik, V., Bousquet, O., & Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning*, *46*, 131–159.

Chapelle, O., & Zien, A. (2005). Semi-supervised classification by low density separation. *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics* (pp. 57–64).

Joachims, T. (1999). Transductive inference for text classification using support vector machines. *Proceedings of the Sixteenth International Conference on Machine Learning* (pp. 41–48).

Lanckriet, G. R. G., Cristianini, N., Ghaoui, L. E., Bartlett, P. L., & Jordan, M. I. (2004). Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, *5*, 27–72.

Micchelli, C., & Pontil, M. (2005). Learning the kernel function via regularization. *Journal of Machine Learning Research*, *6*, 1099–1125.

Rockafellar, R. (1996). *Convex analysis*. Princeton Landmarks in Mathematics and Physics. Princeton University Press; Reprint edition.

Sindhwani, V., Keerthi, S., & Chapelle, O. (2006). Deterministic annealing for semi-supervised kernel machines. *Proceedings of the Twenty-Third International Conference on Machine Learning*.

Sindhwani, V., Niyogi, P., & Belkin, M. (2005). Beyond the point cloud: from transductive to semi-supervised learning. *Proceedings of the Twenty-Second International Conference on Machine Learning*.

Zhang, T., & Ando, R. (2005). *Graph based semi-supervised learning and spectral kernel design* (Technical Report RC23713). IBM T.J. Watson Research Center.

Zhou, D., Bousquet, O., Lal, T., Weston, J., & Schölkopf, B. (2004). Learning with local and global consistency. *Advances in Neural Information Processing Systems 16* (pp. 321–328). MIT Press, Cambridge, MA, USA.

Zhou, D., Huang, J., & Scholkopf, B. (2005). Learning from labeled and unlabeled data on a directed graph. *Proceedings of the Twenty-Second International Conference on Machine Learning* (pp. 1041–1048).

Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. *Proceedings of the Twentieth International Conference on Machine Learning* (pp. 912–919).

Zhu, X., Kandola, J., Ghahramani, Z., & Lafferty, J. (2005). Nonparametric transforms of graph kernels for semi-supervised learning. *Advances in Neural Information Processing Systems 17*.