Semi-supervised Learning Based on Semiparametric Regularization

Zhen Guo* Zhongfei (Mark) Zhang* Eric P. Xing† Christos Faloutsos †

Abstract

Semi-supervised learning plays an important role in the recent literature on machine learning and data mining and the developed semi-supervised learning techniques have led to many data mining applications in recent years. This paper addresses the semi-supervised learning problem by developing a semiparametric regularization based approach, which attempts to discover the marginal distribution of the data to learn the parametric function through exploiting the geometric distribution of the data. This learned parametric function can then be incorporated into the supervised learning on the available labeled data as the prior knowledge. Specifically, our contributions are: (1) We present a semi-supervised learning approach which incorporates the unlabeled data into the supervised learning by a parametric function learned from the whole data including the labeled and unlabeled data. The parametric function reflects the geometric structure of the marginal distribution of the data. Furthermore, the proposed approach which naturally extends to the out-of-sample data is an inductive learning method in nature. (2) This approach allows a family of algorithms to be developed based on various choices of the original RKHS and the loss function. (3) We provide experimental comparisons showing that the proposed approach leads the state-of-the-art performance on a variety of classification tasks. In particular, we demonstrate that this approach can be used successfully in both transductive and semi-supervised settings.

1 Introduction

Semi-supervised learning attempts to use the unlabeled data to improve the performance. The labeled data are often expensive to obtain since they require the efforts of experienced experts. Meanwhile, the unlabeled data are relatively easy to collect. Semi-supervised learning has attracted considerable attention in recent years and many methods have been proposed to utilize the unlabeled data. Most of the semi-supervised learning models are based on the *cluster assumption* which states that the decision boundary should not cross the high density regions, but instead lie in the low density regions. In other words, similar data points should have the same label and dissimilar data points should have different labels.

The approach proposed in this paper is also based on the cluster assumption. Moreover, we believe that the marginal distribution of the data is determined by the unlabeled examples if there is a small labeled data set available along with a relatively large unlabeled data set, which is the case for many applications. The geometry of the marginal distribution must be considered such that the learned classification or regression function adapts to the data distribution. An example is shown in Fig. 1 for a binary classification problem. In Fig. 1(a), the decision function is learned only from the labeled data and the unlabeled data are not used at all. Since the labeled data set is very small, the decision function learned cannot reflect the overall distribution of the data. On the other hand, the marginal distribution of the data described by the unlabeled data has a particular geometric structure. Incorporating this geometric structure into the learning process results in a better classification function, as shown in Fig. 1(b).

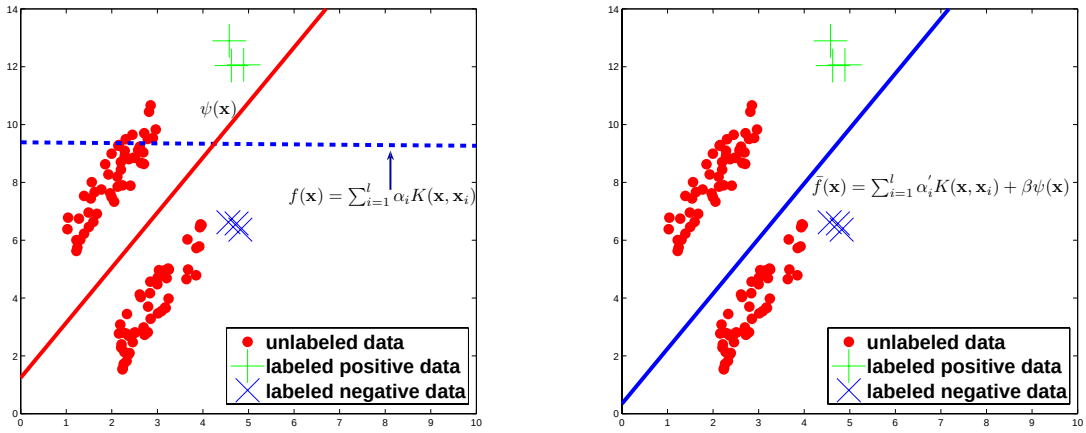
The above observation suggests that the unlabeled data help change the decision function towards the desired direction. Therefore, the question we set for ourselves in this paper is the following:

How to incorporate the geometric structure of the marginal distribution of the data into the learning such that the resulting decision function $\bar{f}$ reflects the distribution of the data?

A variety of graph based methods are proposed in the literature to achieve this goal. The approach presented in this paper exploits the geometric structure in a different way. This is achieved by a 2-step learning process. The first step is to obtain a parametric function from the unlabeled data which describes the geometric structure of the marginal distribution. In this paper, this parametric function is obtained by applying Kernel Principal Component Analysis (KPCA) algorithm to the whole data including the labeled and unlabeled data. In KPCA, the function to extract the most important principal component is a linear combination of the kernel functions in the Reproducing Kernel Hilbert Space (RKHS), $f(\mathbf{x}) = K(\mathbf{x}, .)\boldsymbol{\alpha}$, where K is a kernel function and $\boldsymbol{\alpha}$ is the coefficients vector. This learned parametric function can be shown to reflect the geometric structure of the marginal distribution of the data. The second step is a supervised learning on the labeled data. To incorporate this parametric function into the supervised learning, we extend the original RKHS to be used in the supervised learning by including this parametric function learned from the whole

*Department of Computer Science, SUNY at Binghamton, Binghamton, NY 13902

†School of Computer Science, CMU, Pittsburgh, PA, 15213



(a) (b)

Figure 1: (a) The decision function (dashed line) learned only from the labeled data. (b) The decision function (solid line) learned after the unlabeled data are considered also.

data. Consequently, we call this approach a *semiparametric regularization* based semi-supervised learning.

By selecting different loss functions for the supervised learning, we obtain different semi-supervised learning frameworks. We primarily focus on two families of the algorithms: the semiparametric regularized Least Squares (hereafter SpRLS) and the semiparametric regularized Support Vector Machines (hereafter SpSVM). These algorithms demonstrate the state-of-the-art performance on a variety of classification tasks.

We highlight the following aspects of this paper:

1. We present a semi-supervised learning approach which incorporates the unlabeled data into the supervised learning by a parametric function learned from the whole data including the labeled and unlabeled data. This parametric function reflects the geometric structure of the marginal distribution of the data. Furthermore, the proposed approach which naturally extends to the out-of-sample data is an inductive learning method in nature.

2. This approach allows a family of algorithms to be developed based on various choices of the original RKHS and the loss function.

3. We provide experimental comparisons showing that the proposed approach leads the state-of-the-art performance on a variety of classification tasks. In particular, we demonstrate that this approach can be used successfully in both transductive and semi-supervised settings.

The paper is organized as follows. We first discuss some related work in Section 2. We introduce our approach by reviewing the supervised learning which minimizes the regularized risk functional in RKHS assuming the labeled data only in Section 3. We subsequently introduce the unlabeled data, define the semiparametric regularization, and formulate the semiparametric semi-supervised learning algorithms for different loss functions in Section 4. In Section 5 we then report the results of the experiments where our approach demonstrates the state-of-the-art performance on a variety of classification tasks. The paper concludes in Section 6.

2 Related Work

The idea of regularization has a rich mathematical history dating back to Tikhonov [15] where it is used for solving ill-posed inverse problems. Many machine learning algorithms, including SVM, can be interpreted as examples of regularization. Many existing semi-supervised learning methods rely on the cluster assumption directly or indirectly and exploit the regularization principle by considering additional regularization terms on the unlabeled data. Zhu [20] has an excellent literature survey on the semi-supervised learning. TSVM [16] may be considered as SVM with an additional regularization term on the unlabeled data. Xu et al. [17] propose a TSVM training method based on semi-definite programming. Szummer et al. [14] propose an information regularization framework to minimize the mutual information on multiple overlapping regions covering the data space. The idea is that labels should not change too much in a high density region. Chapelle et al. [6] exploit the same principle. Grandvalet et al. [7] use the entropy on the unlabeled data as

a regularizer. These methods implement the cluster assumption indirectly.

Graph-based methods [3, 21, 9, 5, 13, 19, 8, 18, 11] assume the label smoothness constraint over a graph where the nodes represent the labeled and unlabeled examples and the edges reflect the similarities of the examples. Belkin et al. [2] propose a data-dependent manifold regularization term approximated on the basis of the labeled and unlabeled data using the graph associated with the data. In their approach, the geometric structure of the marginal distribution is extracted using the graph Laplacian associated with the data. In our approach, the geometric structure is described by a parametric function obtained from the whole data including the labeled and unlabeled data. In our 2-step learning process, the classification function we obtain has the same form as that in [2] if we use the same kernel. However, we use different methods to obtain the coefficients. We will discuss this in detail later.

Kernel methods [12, 16] have been widely used in the machine learning community. The semi-supervised learning on the kernel methods becomes very popular in recent years [2, 1, 13, 10]. Sindhwani et al. [13] give a data-dependent non-parametric kernel. They propose to warp an RKHS to adapt to the geometry of the data and derive a modified kernel defined in the same space of the functions as the original RKHS, but with a different norm. Building on [13], Altun et al. [1] propose a graph-based semi-supervised learning framework for structured variables. In this paper, we warp an RKHS in a different way. We extend the original RKHS to be used in the supervised learning by including a parametric function learned from the whole data such that the learned decision function reflects the data distribution. In some cases, this parametric function belongs to the original RKHS and thus the RKHS is not changed. However, the learned classification function still reflects the data distribution. This will be discussed in detail later.

3 Supervised Learning

We begin with the brief review of the supervised learning. Suppose that there is a probability distribution P on $\mathcal{X} \times \mathcal{Y}, \mathcal{X} \subset \mathbb{R}^n$ according to which data are generated. We assume that the given data consist of l labeled data points $(\mathbf{x}_i, y_i), 1 \leq i \leq l$ which are generated according to P. In this paper, we assume the binary classification problem where the labels $y_i, 1 \leq i \leq l$, are binary, i.e., $y_i = \pm 1$.

In the supervised learning scenario, the goal is to learn a function f to minimize the expected loss called risk functional

$$R(f) = \int L(\mathbf{x}, y, f(\mathbf{x}))\, dP(\mathbf{x}, y) \tag{3.1}$$

where L is a loss function. A variety of loss functions have been considered in the literature. The simplest loss function is 0/1 loss

$$L(\mathbf{x}_i, y_i, f(\mathbf{x}_i)) = \begin{cases} 0 & \text{if } y_i = f(\mathbf{x}_i) \\ 1 & \text{if } y_i \neq f(\mathbf{x}_i) \end{cases} \tag{3.2}$$

In Regularized Least Square (RLS), the loss function is given by

$$L(\mathbf{x}_i, y_i, f(\mathbf{x}_i)) = (y_i - f(\mathbf{x}_i))^2$$

In SVM, the loss function is given by

$$L(\mathbf{x}_i, y_i, f(\mathbf{x}_i)) = \max(0, 1 - y_i f(\mathbf{x}_i))$$

For the loss function Eq. (3.2), Eq. (3.1) determines the probability of a classification error for any decision function f. In most applications the probability distribution P is unknown. The problem, therefore, is to minimize the risk functional when the probability distribution function $P(\mathbf{x}, y)$ is unknown but the labeled data $(\mathbf{x}_i, y_i), 1 \leq i \leq l$ are given. Thus, we need consider the empirical estimate of the risk functional [16]

$$R_{emp}(f) = C \sum_{i=1}^{l} L(\mathbf{x}_i, y_i, f(\mathbf{x}_i)) \tag{3.3}$$

where $C > 0$ is a constant. We often use $C = \frac{1}{l}$. Minimizing the empirical risk Eq. (3.3) may lead to numerical instabilities and bad generalization performance [12]. A possible way to avoid this problem is to add a stabilization (regularization) term $\Theta(f)$ to the empirical risk functional. This leads to a better conditioning of the problem. Thus, we consider the following regularized risk functional

$$R_{reg}(f) = R_{emp}(f) + \gamma \Theta(f)$$

where $\gamma > 0$ is the regularization parameter which specifies the tradeoff between minimization of $R_{emp}(f)$ and the smoothness or simplicity enforced by small $\Theta(f)$. A choice of $\Theta(f)$ is the norm of the RKHS representation of the feature space

$$\Theta(f) = \|f\|_K^2$$

where $\|.\|_K$ is the norm in the RKHS $\mathcal{H}_K$ associated with the kernel K. Therefore, the goal is to learn the function f which minimizes the regularized risk functional

$$f^* = \arg\min_{f \in \mathcal{H}_K} C \sum_{i=1}^{l} L(\mathbf{x}_i, y_i, f(\mathbf{x}_i)) + \gamma \|f\|_K^2 \tag{3.4}$$

The solution to Eq. (3.4) is determined by the loss function L and the kernel K. A variety of kernels have been considered in the literature. Three most commonly-used kernel functions are listed in the Table 1 where $\sigma > 0, \kappa > 0, \vartheta < 0$. The following classic Representer Theorem [12] states that the solution to the minimization problem Eq. (3.4) exists in $\mathcal{H}_K$ and gives the explicit form of a minimizer.

THEOREM 3.1. *Denote by* $\Omega : [0,\infty) \to \mathbb{R}$ *a strictly monotonic increasing function, by* $\mathcal{X}$ *a set, and by* $\Lambda : (\mathcal{X} \times \mathbb{R}^2)^l \to \mathbb{R} \cup \{\infty\}$ *an arbitrary loss function. Then each minimizer* $f \in \mathcal{H}_K$ *of the regularized risk*

$$\Lambda((\mathbf{x}_1, y_1, f(\mathbf{x}_1)), \cdots, (\mathbf{x}_l, y_l, f(\mathbf{x}_l))) + \Omega(\|f\|_K)$$

admits a representation of the form

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i K(\mathbf{x}_i, \mathbf{x}) \tag{3.5}$$

with $\alpha_i \in \mathbb{R}$.

According to Theorem 3.1, we can use any regularizer in addition to $\gamma\|f\|_K^2$ which is a strictly monotonic increasing function of $\|f\|_K$. This allows us in principle to design different algorithms. In this paper, we take the simplest approach to use the regularizer $\Omega(\|f\|_K) = \gamma\|f\|_K^2$. Given the loss function L and the kernel K, we substitute Eq. (3.5) into Eq. (3.4) to obtain a minimization problem of the variables $\alpha_i, 1 \le i \le l$. The decision function f^* is immediately obtained from the solution to this minimization problem.

4 Semi-supervised Learning

In the semi-supervised learning scenario, in addition to l labeled data points $(\mathbf{x}_i, y_i), 1 \le i \le l$ we are given u unlabeled data points $\mathbf{x}_i, l+1 \le i \le l+u$ which are drawn according to the marginal distribution $P_{\mathcal{X}}$ of P. The decision function is learned from both the labeled data and the unlabeled data. The semi-supervised learning attempts to incorporate the unlabeled data into the supervised learning in different ways. This paper presents a semi-supervised learning approach based on *semiparametric regularization* which extends the original RKHS by including a parametric function learned from the whole data including the labeled and unlabeled data.

4.1 Semiparametric Regularization In the supervised learning, we may have additional prior knowledge about the solution in many applications. In particular, we may know that a specific parametric component is very likely to be a part of the solution. Or we might want to correct the data for some (e.g., linear) trends to avoid the overfitting. The overfitting degrades the generalization performance when there are outliers.

Table 1: most commonly-used kernel functions

kernel name	kernel function
polynomial kernel	$K(\mathbf{x}, \mathbf{x}_i) = (\langle \mathbf{x}, \mathbf{x}_i \rangle + c)^d$
Gaussian radial basis function kernel	$K(\mathbf{x}, \mathbf{x}_i) = \exp(-\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{2\sigma^2})$
sigmoid kernel	$K(\mathbf{x}, \mathbf{x}_i) = \tanh(\kappa\langle \mathbf{x}, \mathbf{x}_i \rangle + \vartheta)$

Suppose that this additional prior knowledge is described as a family of parametric functions $\{\psi_p\}_{p=1}^M : \mathcal{X} \to \mathbb{R}$. These parametric functions may be incorporated into the supervised learning in different ways. In this paper we consider the following regularized risk functional

$$\bar{f}^* = \arg\min_{\bar{f}} C \sum_{i=1}^{l} L(\mathbf{x}_i, y_i, \bar{f}(\mathbf{x}_i)) + \gamma\|f\|_K^2 \tag{4.6}$$

where $\bar{f} := f + h$ with $f \in \mathcal{H}_K$ and $h \in span\{\psi_p\}$. Consequently, we extend the original RKHS $\mathcal{H}_K$ by including a family of parametric functions ψ_p without changing the norm. The semiparametric representer theorem [12] tells us the explicit form of the solution to Eq. (4.6). The following semiparametric representer theorem is an immediate extension of Theorem 3.1.

THEOREM 4.1. *Suppose that in addition to the assumptions of Theorem 3.1 we are given a set of* M *real valued functions* $\{\psi_p\}_{p=1}^M : \mathcal{X} \to \mathbb{R}$*, with the property that the* $l \times M$ *matrix* $(\psi_p(\mathbf{x}_i))_{ip}$ *has rank* M*. Then for any* $\bar{f} := f + h$ *with* $f \in \mathcal{H}_K$ *and* $h \in span\{\psi_p\}$*, minimizing the regularized risk*

$$\Lambda((\mathbf{x}_1, y_1, \bar{f}(\mathbf{x}_1)), \cdots, (\mathbf{x}_l, y_l, \bar{f}(\mathbf{x}_l))) + \Omega(\|f\|_K)$$

admits a representation of the form

$$\bar{f}(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i K(\mathbf{x}_i, \mathbf{x}) + \sum_{p=1}^{M} \beta_p \psi_p(\mathbf{x}) \tag{4.7}$$

with $\alpha_i, \beta_p \in \mathbb{R}$.

In Theorem 4.1, the parametric functions $\{\psi_p\}_{p=1}^M$ can be any functions. The simplest parametric function is the constant function $\psi_1(\mathbf{x}) = 1, M = 1$ as in the standard SVM model where the constant function is used to maximize the margin.

In Eq. (4.6), the family of parametric functions $\{\psi_p\}_{p=1}^M$ do not contribute to the standard regularizer $\|f\|_K^2$. This need not be a major concern if M is sufficiently smaller than l. In this paper, we use $M = 1$ and this parametric function is learned from the whole data including the labeled and unlabeled data. Therefore, the $l \times M$ matrix $(\psi_p(\mathbf{x}_i))_{ip}$ is a vector whose rank is 1. We denote by $\psi(\mathbf{x})$ this parametric function and by β the corresponding coefficient. Thus, the minimizer of Eq. (4.6) is

$$\bar{f}^*(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + \beta^* \psi(\mathbf{x}) \tag{4.8}$$

where K is the kernel in the original RKHS $\mathcal{H}_K$.

4.2 Learning Parametric Function $\psi(\mathbf{x})$ is obtained by applying the KPCA algorithm [12] to the whole data set. KPCA finds the principal axes in the feature space which carry more variance than any other directions by diagonalizing the covariance matrix $\mathbf{C} = \frac{1}{l+u}\sum_{j=1}^{l+u}\Phi(\mathbf{x}_j)\Phi(\mathbf{x}_j)^\top$, where Φ is a mapping function in the RKHS. To find the principal axes, we solve the eigenvalue problem, $(l+u)\lambda\boldsymbol{\gamma} = K_u\boldsymbol{\gamma}$, where K_u is the kernel used. Let λ denote the largest eigenvalue of K_u and $\boldsymbol{\gamma}$ the corresponding eigenvector. Then the most important principal axis is given by

$$\mathbf{v} = \sum_{i=1}^{l+u}\gamma_i\Phi(\mathbf{x}_i) \tag{4.9}$$

Usually we normalize $\mathbf{v}$ such that $\|\mathbf{v}\| = 1$. Given the data point $\mathbf{x}$, the projection onto the principal axis is given by $\langle\Phi(\mathbf{x}), \mathbf{v}\rangle$. Let $\psi(\mathbf{x}) = \langle\Phi(\mathbf{x}), \mathbf{v}\rangle = K_u(\mathbf{x}, .)\boldsymbol{\gamma}$. Fig. 2 shows an illustrative example for the binary classification problem. As shown in this example, $\psi(\mathbf{x})$ might not be the desired classification function. However, $\psi(\mathbf{x})$ is parallel to the desired classification function (the dashed line). They are different up to a constant. Therefore, $\psi(\mathbf{x})$ reflects the geometric structure of the distribution of the data. From this example, it is clear that the data points projected onto the most important principal axis still keep the original neighborhood relationship. In other words, after projection on the principal axis, similar data points stay close and dissimilar data points are kept far away from each other. In the ideal case of separable binary class problem, we have the following theorem which says that the similar data points in the feature space are still similar to each other after projected on the principal axis.

THEOREM 4.2. *Denote by $\mathcal{C}_i, i = 0, 1$ the set of the data points of each class in the binary class problem. Suppose $\mathcal{C}_i = \{\mathbf{x}|\,\|\Phi(\mathbf{x}) - \mathbf{c}_i\| \le r_i\}$ and $\|\mathbf{c}_0 - \mathbf{c}_1\| > r_0 + r_1$. For each class, suppose that the data points are uniformly distributed in the sphere of radius r_i. $\|.\|$ denotes the Euclidean norm and $\mathbf{v}$ denotes the principal axis derived from KPCA as defined in Eq. (4.9). Then*

$$\forall\mathbf{p} \in \mathcal{C}_i, \mathbf{v}^\top\Phi(\mathbf{p}) \in\ R_i, i = 0, 1$$

where $R_i = [\mu_i - r_i, \mu_i + r_i]$ and $\mu_i = \mathbf{v}^\top\mathbf{c}_i$. Moreover, R_0 and R_1 do not overlap.

Proof. Suppose that the number of the data points in the class $\mathcal{C}_i$ is n_i, respectively. Any data point in the class $\mathcal{C}_i$ can be expressed as $\Phi(\mathbf{x}) = \mathbf{c}_i + r_i\mathbf{t}$ where $\|\mathbf{t}\| \le 1$. Denote by y the projection on the principal axis, $y = \mathbf{v}^\top\Phi(\mathbf{x})$. Therefore, $y = \mathbf{v}^\top\mathbf{c}_i + r_i\mathbf{v}^\top\mathbf{t}$. Since $\|\mathbf{v}\| = 1$, $|\mathbf{v}^\top\mathbf{t}| \le 1$. Thus, the range of y in the class $\mathcal{C}_i$ is $[\mu_i - r_i, \mu_i + r_i]$. Because the sphere is symmetric and the data points are uniformly distributed, the mean of y in the class $\mathcal{C}_i$ is μ_i. Denote by $\delta_i, i = 0, 1$, the variance of y in each class. Note δ_i is invariant to the projection direction. The reason is again that the sphere is symmetric and the data points are uniformly distributed.

Therefore, the overall mean of all y is $\mu = \frac{n_0\mu_0 + n_1\mu_1}{n_0+n_1}$ and the overall variance is

$$\begin{aligned}
\delta = & \frac{1}{n_0+n_1}\sum_y(y-\mu)^2 \\
= & \frac{1}{n_0+n_1}[\sum_{y\in\mathcal{C}_0}(y-\mu)^2 + \sum_{y\in\mathcal{C}_1}(y-\mu)^2] \\
= & \frac{1}{n_0+n_1}[\sum_{y\in\mathcal{C}_0}(y-\mu_0+\frac{n_1}{n_0+n_1}(\mu_0-\mu_1))^2 \\
& + \sum_{y\in\mathcal{C}_1}(y-\mu_1+\frac{n_0}{n_0+n_1}(\mu_1-\mu_0))^2] \\
= & \frac{n_0}{n_0+n_1}\delta_0 + \frac{n_1}{n_0+n_1}\delta_1 + \frac{n_0\,n_1}{(n_0+n_1)^2}(\mu_1-\mu_0)^2
\end{aligned}$$

It can be shown that the ranges of y of the two classes on the principal axis derived from the KPCA do not overlap. First of all, there exists a projection axis such that these two ranges do not overlap. Conceptually, consider the projection axis $\frac{\mathbf{c}_1-\mathbf{c}_0}{\|\mathbf{c}_1-\mathbf{c}_0\|}$. Then we have $\tilde{\mu}_0 = \frac{1}{\|\mathbf{c}_1-\mathbf{c}_0\|}(\mathbf{c}_1-\mathbf{c}_0)^\top\mathbf{c}_0$ and $\tilde{\mu}_1 = \frac{1}{\|\mathbf{c}_1-\mathbf{c}_0\|}(\mathbf{c}_1-\mathbf{c}_0)^\top\mathbf{c}_1$. Thus, $\tilde{\mu}_1 - \tilde{\mu}_0 = \|\mathbf{c}_1 - \mathbf{c}_0\| > r_0 + r_1$. Therefore, these two ranges do not overlap. Denote by $\tilde{\delta}$ the variance in this case. Next we give a formal proof below by contradiction.

Suppose that these two ranges were to overlap under the principal axis derived from the KPCA. Thus, $\|\mu_1 - \mu_0\| < r_0 + r_1$. Consequently, $\delta < \tilde{\delta}$ since δ_0, δ_1 are invariant to the projection axis. This is a contradiction since the variance on the principal axis derived from the KPCA should be the maximum among all the projection axes. Hence, these two ranges do not overlap on the principal axis $\mathbf{v}$ derived from the KPCA.

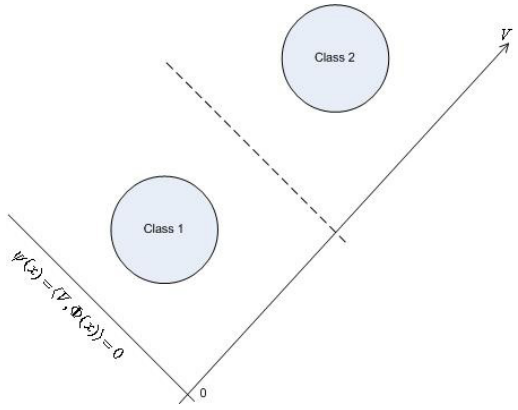


Figure 2: Illustration of KPCA in the two dimensions

Based on the above analysis, our semi-supervised learning is achieved by a 2-step learning process. The first step is to obtain a parametric function $\psi(\mathbf{x})$ from the whole data. Since this parametric function $\psi(\mathbf{x})$ is obtained by KPCA, $\psi(\mathbf{x})$ reflects the geometric structure of the marginal distribution of the data revealed by the whole data. This implements cluster assumption indirectly. The second step is to solve Eq. (4.6) on a new function space to obtain the final classification function.

If $K_u = K$, the final classification function has the form $\bar{f}(\mathbf{x}) = \sum_{i=1}^{l+u} \alpha_i^{'} K(\mathbf{x}_i, \mathbf{x})$ where $\alpha_i^{'}$ is the linear combination of α_i and β. This classification function has the same form as that in [2]. But the methods to obtain it are different. In this case, the parametric function belongs to the original RKHS. Adding $\psi(\mathbf{x})$ does not change the RKHS, but guides the learned classification function towards the desired direction described by $\psi(\mathbf{x})$. If K_u and K are two different kernels, the original RKHS is extended by $\psi(\mathbf{x})$.

The coefficient β^* reflects the weight of the unlabeled data in the learning process. When $\beta^* = 0$, the unlabeled data are not considered at all and this method is a fully supervised learning algorithm. This means that the unlabeled data do not provide any useful information. In other words, the unlabeled data follow the marginal distribution described by the labeled data. When $\beta^* \neq 0$, the unlabeled data provide the useful information about the marginal distribution of the data and the geometric structure of the marginal distribution revealed by the unlabeled data is incorporated into the learning.

To learn the final classification function, we substitute Eq. (4.8) into Eq. (4.6) to obtain an objective function of α_i^* and β^*. The solution of α_i^* and β^* depends on the loss function. Different loss functions L result in different algorithms. We now discuss two typical loss functions: the squared loss for RLS and the hinge loss for SVM. For the squared loss function, we obtain the explicit form of α_i^* and β^*. In the following sections, we use K interchangeably to denote the kernel function or the kernel matrix.

4.3 Semiparametric Regularized Least Squares We first outline the RLS approach which applies to the binary classification and the regression problem. The classic RLS algorithm is a supervised method where we solve:

$$f^* = \arg\min_{f \in \mathcal{H}_K} C \sum_{i=1}^{l} (y_i - f(\mathbf{x}_i))^2 + \gamma \|f\|_K^2$$

where C and γ are the constants.

According to Theorem 3.1, the solution is of the following form

$$f^*(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i^* K(\mathbf{x}_i, \mathbf{x})$$

Substituting this solution in the problem above, we arrive at the following differentiable objective function of the l-dimensional variable $\boldsymbol{\alpha} = [\alpha_1 \cdots \alpha_l]^\top$:

$$\boldsymbol{\alpha}^* = \arg\min C(Y - K\boldsymbol{\alpha})^\top (Y - K\boldsymbol{\alpha}) + \gamma \boldsymbol{\alpha}^\top K \boldsymbol{\alpha}$$

where K is the $l \times l$ kernel matrix $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{Y}$ is the label vector $\mathbf{Y} = [y_1 \cdots y_l]^\top$.

The derivative of the objective function over $\boldsymbol{\alpha}$ vanishes at the minimizer

$$C(KK\boldsymbol{\alpha}^* - K\mathbf{Y}) + \gamma K \boldsymbol{\alpha}^* = 0$$

which leads to the following solution.

$$\boldsymbol{\alpha}^* = (CK + \gamma \mathbf{I})^{-1} C\mathbf{Y}$$

The semiparametric RLS algorithm solves the optimization problem in Eq. (4.6) with the squared loss function:

$$\bar{f}^* = \arg\min_{\bar{f}} C \sum_{i=1}^{l} (y_i - \bar{f}(\mathbf{x}_i))^2 + \gamma \|f\|_K^2 \tag{4.10}$$

where $\bar{f} := f + h$ with $f \in \mathcal{H}_K$ and $h \in span\{\psi\}$.

According to Theorem 4.1, the solution has the form of

$$\bar{f}^* = \sum_{i=1}^{l} \alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + \beta^* \psi(\mathbf{x})$$

Substituting this form in Eq. (4.10), we arrive at the following objective function of the l-dimensional variable $\boldsymbol{\alpha} = [\alpha_1 \cdots \alpha_l]^\top$ and β:

$$(\boldsymbol{\alpha}^*, \beta^*) = \arg\min C\boldsymbol{\delta}^\top \boldsymbol{\delta} + \gamma \boldsymbol{\alpha}^\top K \boldsymbol{\alpha}$$

where $\boldsymbol{\delta} = Y - K\boldsymbol{\alpha} - \beta\boldsymbol{\psi}$, K is the $l \times l$ kernel matrix $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{Y}$ is the label vector $\mathbf{Y} = [y_1 \cdots y_l]^\top$, and $\boldsymbol{\psi}$ is the vector $\boldsymbol{\psi} = [\psi(\mathbf{x}_1) \cdots \psi(\mathbf{x}_l)]^\top$. The derivatives of the objective function over $\boldsymbol{\alpha}$ and β vanish at the minimizer:

$$\begin{aligned} C(KK\boldsymbol{\alpha}^* + \beta^* K\boldsymbol{\psi} - K\mathbf{Y}) + \gamma K\boldsymbol{\alpha}^* &= 0 \\ \boldsymbol{\psi}^\top K\boldsymbol{\alpha}^* + \beta^* \boldsymbol{\psi}^\top \boldsymbol{\psi} - \boldsymbol{\psi}^\top Y &= 0 \end{aligned}$$

which lead to the following solution:

$$\begin{aligned} \boldsymbol{\alpha}^* &= C(\gamma \mathbf{I} - \frac{C\boldsymbol{\psi}\boldsymbol{\psi}^\top K}{\boldsymbol{\psi}^\top \boldsymbol{\psi}} + CK)^{-1} (\mathbf{I} - \frac{\boldsymbol{\psi}\boldsymbol{\psi}^\top}{\boldsymbol{\psi}^\top \boldsymbol{\psi}}) \mathbf{Y} \\ \beta^* &= \frac{\boldsymbol{\psi}^\top Y - \boldsymbol{\psi}^\top K \boldsymbol{\alpha}^*}{\boldsymbol{\psi}^\top \boldsymbol{\psi}} \end{aligned} \tag{4.11}$$

4.4 Semiparametric Regularized Support Vector Machines We outline the SVM approach to the binary classification problem which is the focus of this paper. In the binary

classification problem, the classic SVM attempts to solve the following optimization problem on the labeled data.

$$\begin{aligned}(4.12)\qquad \min \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\xi_i \\ s.t. \quad & y_i\{\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b\} \geq 1 - \xi_i \\ & \xi_i \geq 0 \quad i = 1, \cdots, l\end{aligned}$$

where Φ is a nonlinear mapping function determined by the kernel and b is a regularized term.

Again, the solution is given by

$$f^*(\mathbf{x}) = \langle \mathbf{w}^*, \Phi(\mathbf{x})\rangle + b^* = \sum_{i=1}^{l}\alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + b^*$$

To solve Eq. (4.12) we introduce one Lagrange multiplier for each constraint in Eq. (4.12) using the Lagrange multipliers technique and obtain a quadratic dual problem of the Lagrange multipliers.

$$\begin{aligned}(4.13)\ \min \quad & \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \mu_i \mu_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{l}\mu_i \\ s.t. \quad & \sum_{i=1}^{l}\mu_i y_i = 0 \\ & 0 \leq \mu_i \leq C \quad i = 1, \cdots, l\end{aligned}$$

where μ_i is the Lagrange multiplier associated with the i-th constraint in Eq. (4.12).

We have $\mathbf{w}^* = \sum_{i=1}^{l}\mu_i y_i \Phi(\mathbf{x}_i)$ from the solution to Eq. (4.13). Note that the following conditions must be satisfied according to the Kuhn-Tucker theorem [16]:

$$(4.14)\ \mu_i(y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b) + \xi_i - 1) = 0 \quad i = 1, \cdots, l$$

The optimal solution of b is determined by the above conditions.

Therefore, the solution is given by

$$f^*(\mathbf{x}) = \sum_{i=1}^{l}\alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + b^*$$

where $\alpha_i^* = \mu_i y_i$.

The semiparametric SVM algorithm solves the optimization problem in Eq. (4.6) with the hinge loss function:

$$\begin{aligned}(4.15)\ \min \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\xi_i \\ s.t. \quad & y_i\{\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b + \beta\psi(\mathbf{x}_i)\} \geq 1 - \xi_i \\ & \xi_i \geq 0 \quad i = 1, \cdots, l\end{aligned}$$

As in the classic SVM, we consider the Lagrange dual problem for Eq. (4.15).

$$\begin{aligned}(4.16)\ \min \quad & \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \mu_i \mu_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{l}\mu_i \\ s.t. \quad & \sum_{i=1}^{l}\mu_i y_i = 0 \\ & \sum_{i=1}^{l}\mu_i y_i \psi(\mathbf{x}_i) = 0 \\ & 0 \leq \mu_i \leq C \quad i = 1, \cdots, l\end{aligned}$$

where μ_i is the Lagrange multiplier associated with the i-th constraint in Eq. (4.15). The semiparametric SVM dual problem Eq. (4.16) is the same as the SVM dual problem Eq. (4.13) except one more constraint introduced by the parametric function $\psi(\mathbf{x})$. As in the classic SVM, the following conditions must be satisfied:

$$(4.17)\quad \mu_i(y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b + \beta\psi(\mathbf{x}_i)) + \xi_i - 1) = 0$$

We have $\mathbf{w}^* = \sum_{i=1}^{l}\mu_i y_i \Phi(\mathbf{x}_i)$ from the solution to Eq. (4.16). This is the same as that in the SVM.

The optimal solution of b^* and β^* is determined by Eq. (4.17). If the number of the Lagrange multipliers satisfying $0 < \mu_i < C$ is no less than two, we may determine b^* and β^* by solving two linear equations corresponding to any two of them in Eq. (4.17) since the corresponding slack variable ξ_i is zero. In the case that the number of the Lagrange multipliers satisfying $0 < \mu_i < C$ is less than two, b^* and β^* are determined by solving the following optimization problem derived from Eq. (4.17).

$$\begin{aligned}(4.18)\qquad \min \quad & b^2 + \beta^2 \\ s.t. \quad & y_i\{\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b + \beta\psi(\mathbf{x}_i)\} \geq 1 \\ & if\ \mu_i = 0 \\ & y_i\{\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b + \beta\psi(\mathbf{x}_i)\} = 1 \\ & if\ 0 < \mu_i < C\end{aligned}$$

The final decision function is

$$\bar{f}^*(\mathbf{x}) = \sum_{i=1}^{l}\alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + \beta^*\psi(\mathbf{x}) + b^*$$

where $\alpha_i^* = \mu_i y_i$. Semiparametric SVM can be implemented by using a standard quadratic programming problem solver.

4.5 Semiparametric Regularization Algorithm Based on the above analysis, the semiparametric regularization algorithm is summarized in Algorithm 1.

Algorithm 1 Semiparametric Regularization Algorithm

Input:
l labeled data points $(\mathbf{x}_i, y_i), 1 \le i \le l, y_i = \pm 1$ and u unlabeled data points $\mathbf{x}_i, l+1 \le i \le l+u$.

Output:
Estimated function $\bar{f}^*(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + \beta^* \psi(\mathbf{x})$ for SpRLS or $\bar{f}^*(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + \beta^* \psi(\mathbf{x}) + b^*$ for SpSVM.

1: **procedure**
2: Choose the kernel K_u and apply KPCA to the whole data to obtain the parametric function $\psi(\mathbf{x}) = \sum_{i=1}^{l+u} \gamma_i K_u(\mathbf{x}_i, \mathbf{x})$.
3: Choose the kernel K and solve Eq. (4.11) for SpRLS or Eqs. (4.16) and (4.18) for SpSVM.
4: **end procedure**

4.6 Transductive Learning and Semi-supervised Learning The transductive learning only works on the labeled and unlabeled training data and cannot handle unseen data. Out-of-sample extension is already a serious limitation for transductive learning. Contrast to the transductive learning, the inductive learning can handle unseen data. The semi-supervised learning can be either transductive or inductive. Many existing graph-based semi-supervised learning methods are transductive in nature since the classification function is only defined on the labeled and unlabeled training data. One reason is that they perform the semi-supervised learning only on the graph where the nodes are the labeled and unlabeled data in the training set, not on the whole space.

In our approach, the decision function Eq. (4.8) is defined over the whole $\mathcal{X}$ space. Therefore, the approach in this paper is inductive in nature and can extend to the out-of-sample data.

4.7 Comparisons with other methods In the literature, many existing semi-supervised learning methods rely on the *cluster assumption* directly or indirectly and exploit the regularization principle by considering additional regularization terms on the unlabeled data. Belkin et al. [2] propose a manifold regularization approach where the geometric structure of the marginal distribution is extracted using the graph Laplacian associated with the data. They considered the following regularization term.

$$\sum_{i,j=1}^{l+u} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 W_{ij} = \mathbf{f}^\top \mathbf{L} \mathbf{f} \tag{4.19}$$

where W_{ij} are edge weights in the data adjacency graph and $\mathbf{L}$ is the graph Laplacian given by $\mathbf{L} = \mathbf{D} - \mathbf{W}$. Here, the diagonal matrix $\mathbf{D}$ is given by $D_{ii} = \sum_{j=1}^{l+u} W_{ij}$. The incorporation of this regularization term leads to the following optimization problem.

$$f^* = \arg\min_{f \in \mathcal{H}_K} C \sum_{i=1}^{l} L(\mathbf{x}_i, y_i, f(\mathbf{x}_i)) + \gamma \|f\|_K^2 + \mathbf{f}^\top \mathbf{L} \mathbf{f}$$

Eq. (4.19) attempts to give the nearby points (large W_{ij}) in the graph similar labels. However, the issue is that Eq. (4.19) tends to give the similar labels for points i and j as long as $W_{ij} > 0$. In other words, dissimilar points might have similar labels. Therefore, their approach depends on the neighborhood graph constructed from the data. Similarly, Zhu et al. [21] minimize Eq. (4.19) as an energy function.

The semiparametric regularization based semi-supervised learning approach in this paper exploits the cluster assumption by the parametric function $\psi(\mathbf{x})$. Learned from the whole data, this parametric function reflects the geometric structure of the marginal distribution of the data. Different from the manifold regularization approach, our approach uses a parametric function obtained from the whole data to describe the geometric structure of the marginal distribution. Similar to the manifold regularization approach, our approach obtains the same form of the classification function if we use the same kernel ($K = K_u$) in the 2-step learning process. However, the methods to obtain the expansion coefficients are different.

Sindhwani et al. [13] derive a modified kernel defined in the same space of the functions as the original RKHS, but with a different norm. In this paper, we warp an RKHS in a different way. We extend the original RKHS by including the parametric function without changing the norm such that the learned decision function reflects the data distribution. In some cases, this parametric function belongs to the original RKHS and thus the RKHS is unchanged. However, the learned classification function still reflects the data distribution since the classification function has a preference to the parametric function according to Eq. (4.8).

The parametric function $\psi(\mathbf{x})$ learned by KPCA can be incorporated into the supervised learning to separate different classes very well for the binary classification problem. For the multiclass problem, KPCA cannot separate different class very well because some classes overlap after projection onto the principal axis. That is why we focus on the binary class problem in this paper.

5 Experiment Results

Experiments are performed on seven well-known datasets described in Table 2 where c is the number of classes, d is the data dimension, l is the number of the labeled data points, and n is the total number of the data points in the dataset including labeled, unlabeled, and test data points. The dataset g50c, mac-win and WebKb are from [13]. The dataset g24lc and BCI are from [4]. g50c is an artificial dataset generated from two unit-covariance normal distributions with

Table 2: Dataset used in the experiments

Dataset	c	d	l	n
g50c	2	50	50	550
g24lc	2	241	50	1500
mac-win	2	7511	50	1946
BCI	2	117	50	400
WebKb(page)	2	3000	12	1051
WebKb(link)	2	1840	12	1051
WebKb(page+link)	2	4840	12	1051

equal probabilities. g24lc is artificially generated such that the cluster assumption holds, but the manifold assumption does not. mac-win is taken from the newsgroups20 dataset and the task is to categorize the newsgroup documents into two topics: *mac* or *windows*. BCI dataset originates from research toward the development of a brain computer interface. The WebKb dataset is a subset of the web documents of the computer science departments of four universities. The two categories are *course* or *non-course*. For each document, there are two representations: the textual content of the webpage (which we call *page* representation) and the anchor text on links on other webpages pointing to the webpage (which we call *link* representation). We also consider a joint ($page + link$) representation by concatenating the features.

We compare SpRLS and SpSVM with the methods in Sindhwani et al. [13] (thus called LapRLS and LapSVM for the reference purpose) as well as the original RLS and SVM in performance. In our experiments, K is set as the same as K_u as the Gaussian RBF kernel. For g50c, mac-win, and WebKb datasets, we use the same kernel parameters as those used in [13] which also uses the Gaussian RBF kernel and chooses the parameters using the cross-validation method. Sindhwani et al. [13] did not report the experimental results on g241c or BCI datasets. Therefore, we choose the kernel parameters based on the performance on a small grid of parameter values and apply the same parameters to the LapSVM and LapRLS algorithms. We choose the regularization parameters (e.g., C in the Eq. (4.15)) based on the performance on a small grid of parameter values, too.

In the transductive setting, the training set consists of n examples, l of which are labeled (n, l are specified in Table 2). Table 3 reports the results for predicting the labels of the $n - l$ unlabeled data points under the transductive setting. The performance is evaluated by the error rates (mean and standard deviation) on the unlabeled data averaged over 10 runs with different random choices of the labeled set.

In the semi-supervised setting, the training set consists of l labeled data points and u unlabeled data points; the test set consists of $n - l - u$ data points. Table 4 reports the results for predicting the labels of the unlabeled data and the test data for g50c, g24lc, mac-win, and BCI datasets. Table 5 reports the results for WebKb dataset. The performance is evaluated again by the error rates averaged over 10 runs with different random choices of the labeled data and the unlabeled data.

In summary, our approach outperforms LapSVM and LapRLS in all the cases in the transductive setting except on the WebKb (page) dataset. In the semi-supervised setting, our approach outperforms LapSVM and LapRLS in all the cases. In both settings, SpRLS returns the best performance and outperforms SpSVM in most cases. One possible reason might be that we use MATLAB to solve the quadratic optimization problem in the SpSVM and MATLAB does not support quadratic optimization very well.

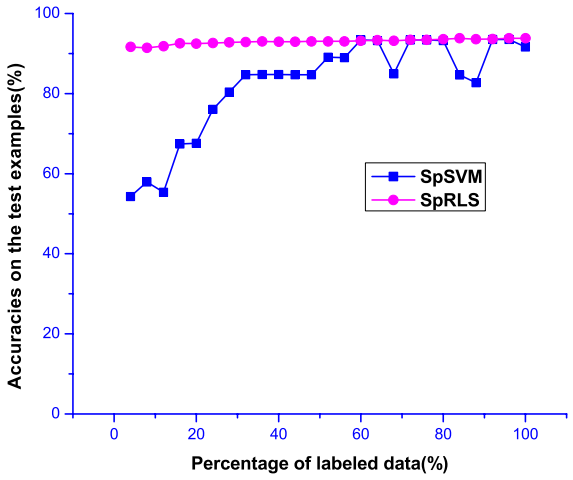


Figure 3: Accuracies on the test data with different percentages of the labeled data for the g50c dataset

We also evaluate the performance in terms of the accuracy on the test data with different percentages of the labeled data in the training set while keeping the size of the whole training set as a constant. We define the performance accuracy as the correct percentage w.r.t the ground truth. Fig. 3 reports the result on the g50c dataset and Fig. 4 reports the result on the mac-win dataset. SpRLS demonstrates a good performance even with a very few labeled data. For g50c dataset, SpRLS only needs two labeled data points (one for each class) to obtain a performance almost as good as that using 100 labeled data points. From this figure, it is clear that as long as we have a sufficiently few labeled data samples ($\geq 2\%$ for the g50c dataset and $\geq 24\%$ for the mac-win dataset), this method ensures a satisfactory classification performance (around 70% accuracy).

Table 3: Transductive setting: Error rates on the unlabeled examples

Dataset→ Algorithm↓	g50c	g24lc	mac-win	BCI	WebKB (link)	WebKB (page)	WebKB (page+link)
SVM(full labels)	8.0(0.4)	6.4(0.1)	2.5(0.1)	29.0(1.4)	12.4(0.1)	13.1(0.1))	10.5(0.1)
RLS(full labels)	2.5(0.1)	0(0)	0(0)	0(0)	0.5(0)	0.6(0)	0.2(0)
LapSVM	6.1(1.1)	35.4(6.8)	10.5(2.0)	49.8(2.0)	20.2(11.4)	13.0(6.8)	15.1(7.4)
LapRLS	5.4(1.1)	34.5(8.5)	10.1(1.4)	49.4(2.3)	31.3(24.8)	**7.9(2.7)**	11.0(7.7)
SpSVM	18.7(21.8)	34.0(29.5)	**7.1(0.7)**	49.6(1.3)	64.3(29.0)	57.4(33.3)	78.1(0.1)
SpRLS	**5.2(0.9)**	**14.8(2.4)**	8.0(1.7)	**37.4(2.5)**	**13.5(4.4)**	10.9(5.9)	**4.3(1.9)**

Table 4: Semi-supervised setting: Error rates on the unlabeled and test examples for g50c, g24lc, mac-win, and BCI datasets

Dataset→	g50c		g24lc		mac-win		BCI	
Algorithm↓	unlabel	test	unlabel	test	unlabel	test	unlabel	test
SVM	11.7(5.7)	9.7(6.0)	48.2(2.1)	48.1(3.2)	45.4(10.2)	47.6(11.4)	49.2(2.1)	49.8(6.8)
RLS	20.6(10.4)	19.4(10.0)	29.6(6.1)	30.4(7.6)	46.5(10.9)	47.4(11.4)	37.9(2.8)	36.7(3.3)
LapSVM	7.2(1.3)	7.0(1.8)	34.4(6.7)	34.9(8.6)	10.8(1.3)	11.1(2.6)	50.2(1.4)	44.9(4.4)
LapRLS	6.4(1.2)	6.2(1.6)	33.2(8.6)	33.1(9.6)	10.1(1.4)	10.5(2.4)	49.1(1.6)	42.4(5.2)
SpSVM	10.3(14.1)	9.8(14.6)	17.7(11.2)	18.9(12.1)	**7.6(1.3)**	9.2(2.4)	48.4(2.7)	50.4(5.6)
SpRLS	**5.5(1.1)**	**4.9(1.7)**	**15.2(2.4)**	**17.1(4.1)**	8.1(1.8)	**9.0(2.7)**	**37.8(2.8)**	**36.7(3.3)**

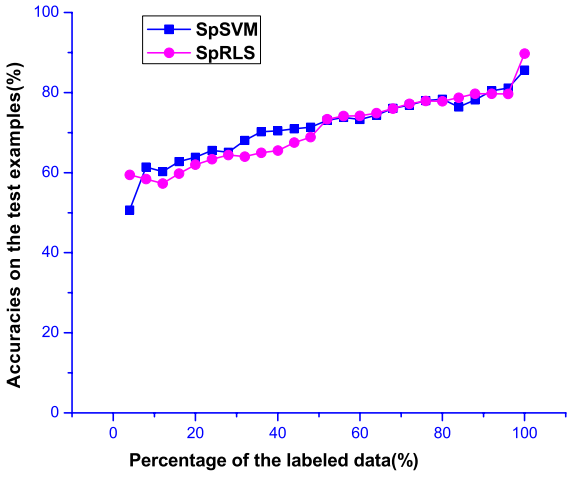


Figure 4: Accuracies on the test data with different percentages of the labeled data for the mac-win dataset

6 Conclusion

This paper presents a semi-supervised learning approach based on semiparametric regularization which extends to the out-of-sample data points. A specific parametric function is learned from the whole data including the plentiful unlabeled data. This specific parametric function is then incorporated into the supervised learning on a few available labeled data to exploit the geometric structure of the marginal distribution of the data. This approach allows a family of algorithms to be developed based on various choices of the original RKHS and the loss function. Empirical evaluations demonstrate that the proposed approach outperforms the state-of-the-art methods in the literature on a variety of classification tasks.

7 Acknowledgement

We thank the anonymous reviewers for insightful comments. This work is supported in part by NSF (IIS-0535162, IIS-0534205, DBI-0640543), AFRL (FA8750-05-2-0284), AFOSR (FA9550-06-1-0327), and NIH (1090151). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] Y. Altun, D. McAllester, and M. Belkin. Maximum margin semi-supervised learning for structured variables. In *Advances in Neural Information Processing Systems (NIPS) 18*, 2005.

Table 5: Semi-supervised setting: Error rates on the unlabeled and test examples for WebKb dataset

Dataset→	WebKb(page)		WebKb(link)		WebKb(page+link)	
Algorithm↓	unlabel	test	unlabel	test	unlabel	test
SVM	27.6(17.8)	27.1(17.3)	20.8(2.3)	19.6(2.5)	20.2(2.7)	19.8(4.7)
RLS	21.9(0.6)	21.7(1.6)	22.2(0.9)	20.6(2.5)	18.7(6.5)	18.8(7.2)
LapSVM	16.4(6.8)	16.4(5.5)	16.1(4.6)	15.1(5.6)	15.7(7.3)	16.4(7.2)
LapRLS	14.2(6.6)	15.0(6.1)	31.4(24.5)	28.7(26.2)	13.2(7.4)	14.7(7.5)
SpSVM	57.5(33.3)	57.6(32.5)	70.6(22.2)	72.1(23.0)	78.0(0.6)	78.5(2.3)
SpRLS	**10.1(4.5)**	**9.7(5.3)**	**13.7(4.4)**	**13.3(4.6)**	**4.2(1.7)**	**5.0(2.6)**

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

[3] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *ICML*, pages 19–26, 2001.

[4] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[5] O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. In *NIPS*, pages 585–592, 2002.

[6] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, 2005.

[7] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *NIPS*, 2004.

[8] T. Joachims. Transductive learning via spectral graph partitioning. In *ICML*, pages 290–297, 2003.

[9] A. Kapoor, Y. A. Qi, H. Ahn, and R. W. Picard. Hyperparameter and kernel learning for graph based semi-supervised classification. In *NIPS*, 2005.

[10] J. D. Lafferty, X. Zhu, and Y. Liu. Kernel conditional random fields: representation and clique selection. In *ICML*, 2004.

[11] T. P. Pham, H. T. Ng, and W. S. Lee. Word sense disambiguation with semi-supervised learning. In *AAAI*, pages 1093–1098, 2005.

[12] B. Schölkopf and A. Smola. *Learning with Kernels Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Cambridge, MA, 2002.

[13] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *Proc. ICML*, 2005.

[14] M. Szummer and T. Jaakkola. Information regularization with partially labeled data. In *Advances in Neural Information Processing Systems, 15*, 2002.

[15] A. N. Tikhonov. On solving ill-posed problem and method of regularization. *Dokl. Akad. Nauk USSR 153*, pages 501–504, 1963.

[16] V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc, 1998.

[17] L. Xu and D. Schuurmans. Unsupervised and semi-supervised multi-class support vector machines. In *AAAI*, pages 904–910, 2005.

[18] X. Zhang and W. S. Lee. Hyperparameter learning for graph based semi-supervised learning algorithms. In *NIPS*, 2006.

[19] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, 2003.

[20] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

[21] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.