Subspace Regularization: A New Semi-supervised Learning Method

Yan-Ming Zhang, Xinwen Hou, Shiming Xiang, and Cheng-Lin Liu

National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences,
95 Zhongguancun East Road, Beijing 100190, P.R. China
{ymzhang,xwhou,smxiang,liucl}@nlpr.ia.ac.cn

Abstract. Most existing semi-supervised learning methods are based on the smoothness assumption that data points in the same high density region should have the same label. This assumption, though works well in many cases, has some limitations. To overcome this problems, we introduce into semi-supervised learning the classic low-dimensionality embedding assumption, stating that most geometric information of high dimensional data is embedded in a low dimensional manifold. Based on this, we formulate the problem of semi-supervised learning as a task of finding a subspace and a decision function on the subspace such that the projected data are well separated and the original geometric information is preserved as much as possible. Under this framework, the optimal subspace and decision function are iteratively found via a projection pursuit procedure. The low computational complexity of the proposed method lends it to applications on large scale data sets. Experimental comparison with some previous semi-supervised learning methods demonstrates the effectiveness of our method.

1 Introduction

We consider the general problem of learning from labeled and unlabeled data. Given an input data set $\{x_1, \ldots, x_l, x_{l+1}, \ldots, x_n\}$, the first l points have labels $\{y_1, \ldots, y_l\} \in \{-1, +1\}$ and the remaining points are unlabeled. The goal is to learn a prediction function which has low classification error on test points.

To make use of unlabeled data, assumption of the relationship between the marginal distribution $p(x)$ and the conditional distribution $p(y|x)$ should be made. This prior assumption plays an essential role in semi-supervised learning [1,2]. Most of the semi-supervised learning methods proposed by far are based on the smoothness assumption that "two points in the same high-density region are likely of the same label" [1]. The effectiveness and generality of this assumption has made the smoothness-based methods very successful, and in fact most of the state-of-the-art semi-supervised learning methods are based on this assumption.

Although the smoothness assumption is effective and methods based on it have obtained good performance on some problems, there are two main limitations of it. First, according to Mann and McCallum [3], most of the current

W. Buntine et al. (Eds.): ECML PKDD 2009, Part II, LNAI 5782, pp. 586–601, 2009.

semi-supervised methods lack in scalability. Typical semi-supervised learning methods, such as label propagation [4,5,2], Manifold Regularization [6], Transductive Support Vector Machines (TSVM) [7], require learning time of order $O(n^3)$ where n is the size of data set. Second, when data of different classes overlap heavily, the ideal decision boundary should cross the overlapping area which is a high density region. Thus, smoothness-based methods may fail since they always avoid a decision boundary that crosses high density region to satisfy the smoothness assumption [3,8].

In this paper, we turn to consider the low-dimensionality embedding assumption in the semi-supervised learning setting. This assumption can be roughly described as most information of high dimensional data is embedded in a low dimensional manifold. It has been traditionally used in dimension reduction methods to map the high dimensional data to a low-dimensional representation while preserving most original information [9].

In the typical setting of semi-supervised learning, there is very limited number of labeled data in a high-dimensional input space. According to statistical learning theory, because of the sparseness of data it is impossible to learn a good classifier for a general problem [10]. Based on the low-dimensional embedding assumption, we naturally hope to find a low-dimensional representation of data so as to make the labeled data more dense and therefore much easier for training. To this end, both labeled and unlabeled data can be used to explore the low-dimensional structure in data. Specifically, we propose a combined criterion to evaluate the candidate subspace and the classifier simultaneously: the data-fitting term evaluates how well the labeled data points of different classes are separated in the subspace, and the regularization term evaluates how much information has been lost by mapping the whole (labeled and unlabeled) data to the subspace. As the regularization term tends to find the subspace that preserves most interesting information, we call this framework as subspace regularization.

Within the above general framework, we instantiate a specific algorithm called PCA-based Least Square (PCA-LS), where the regularization term aims at reducing the reconstruction error of input points just like Principal Component Analysis (PCA) [11]. We also kernelize the PCA-LS to extend our method to nonlinear cases. The method we use to solve the optimal problem turns out to be a special case of the classic projection pursuit procedure which constructs the subspace and the decision function defined on this subspace in an incremental manner.

Compared to the smoothness-based methods, subspace regularization has two remarkable advantages. First, our methods still work when data from different classes overlap heavily, while smoothness-based methods may fail. Roughly speaking, after the subspace is fixed, subspace regularization looks for a decision boundary that can optimally classify the projected labeled data. Thus, although the data still overlap in the subspace, the decision boundary will not be affected by the data density directly and avoids the problem that fails smoothness-based methods. Second, our method has very low computational complexity which implies that it can be applied in large-scale applications. For linear PCA-LS, the computational complexity is linear in the number of data points and dimension of the input space. As a result, for a data set of $80,000$ points, our method is about 60 times faster than typical smoothness-based methods in training. Beside these, the method is rather robust to hyperparameters and still suitable when l/n is large. We will examine these in much more detail in the experiment section.

The remainder of this paper is organized as follows. In section 2, we briefly review the existing semi-supervised learning methods. Section 3 introduces the subspace regularization framework and the details of algorithms. Experimental results are reported in section 4. We conclude the paper in section 5.

2 Related Work

Over the past decade, there are many methods have been proposed to handle the semi-supervised problem. Based on the prior assumptions they use, these methods can be roughly divided into three categories.

The methods from the first category are generative models in which they make a prior assumption on the form of the input data distribution, for example Gaussian mixture models or naive Bayes models [8]. Then, models are trained by Expectation Maximization algorithm using both labeled and unlabeled data. Nigam et al. applied this method to text classification problem and improved the performance dramatically [12]. However, this prior assumption on data distribution is too strict for general problems, and when the model is wrong unlabeled data may hurt accuracy.

The methods from the second category are based on the smoothness assumption as stated in the previous section. Based on this assumption, two families of methods have been developed.

The methods in the first family, namely, graph-based methods, have been developed to satisfy the smoothness assumption by penalizing the variance of decision function in high density region of data. Specifically, using both labeled and unlabeled data, an adjacency graph is constructed to explore the intrinsic geometric structure of the data. The decision function is then found by minimizing the training error on labeled data and the variance on the adjacency graph. Many famous semi-supervised learning methods, like label propagation [4,5,2], spectral methods [13,14], manifold regularization [6], belong to this family.

The methods in the second family, namely, low-density-separation-based methods, implement the smoothness assumption based on an equivalent assumption that "the best decision boundary should be located in low-density region". The aim of this kind of methods is to find a decision boundary which can correctly classify the labeled data and meanwhile is far away from the high density region of unlabeled data points. The distance from one point to the decision boundary is evaluated by the absolute value of the decision function or the value of posterior probability on the point. Methods of this family include the TSVM [7], semi-supervised Gaussian processes [15], entropy regularization [16], information regularization [17], low density separation [18], etc.

The methods from the third category make use of the prior knowledge of label distribution. They force the model predictions on unlabeled data to match the prior label distribution. For example, Zhu et al. used class mean normalization (CMN) as a post-processing step of Gaussian Random Fields (GRF) method to adjust the classification results [4]. Mann et al. proposed a method named Expectation Regularization that directly augments the objective function by adding a regularization term which is defined as the KL-divergence between prior label distribution and empirical label distribution predicted by model [3]. One important advantage of the methods of this type is their high efficiency in computation. However, as they do not explicitly explore the underling structure of data, these methods can not utilize the information of unlabeled data sufficiently.

3 Subspace Regularization

In this section, we first present the subspace regularization framework. Then a specific algorithm is given to learn linear decision function, and kernelized to tackle the nonlinear case. Finally, we analyze the computational complexity of our method.

3.1 Objective Function

Given a set of labeled and unlabeled data, we denote the input data by matrix $X = [x_1, \ldots, x_n]$, and the output by vector $Y = [y_1, \ldots, y_l]^T$. Without confusion, we use W to denote the subspace $W = span\{w_1, \ldots, w_p | w_i \perp w_j, i \neq j\}$ and the matrix $W = [w_1, \ldots, w_p]$ depending on the context.

From the above discussion, we aim to find a low-dimensional subspace $W = span\{w_1, \ldots, w_p | w_i \perp w_j, i \neq j\}$ and a decision function g defined on the W such that the following objective is minimized:

$$L(X, Y, W, g) = \sum_{i=1}^{l} L_F(y_i, g(x_i^T W)) + \lambda L_R(X, X^W), \tag{1}$$

where L_F and L_R are loss functions, and $X^W = [x_1^W, \ldots, x_n^W]$ in which x_i^W is the projection of x_i onto the subspace W. The first term evaluates how well the projected labeled data can be separated by g in the subspace W, and the second term evaluates how much information is lost by projecting data onto the subspace.

Specifically, we choose L_F as the least square error, L_R as the reconstruction error, and let g be an arbitrary linear function defined on W. Then, the objective function can be rewritten as

$$L(X, Y, W, g) = \sum_{i=1}^{l} (y_i - \sum_{t=1}^{p} \alpha_t x_i^T w_t)^2 + \lambda \sum_{i=1}^{n} \|x_i - x_i^W\|^2. \tag{2}$$

The parameters $\alpha = [\alpha_1, ..., \alpha_p]$ and W are estimated by minimizing (2). The dimension of subspace p and the regularization factor λ are hyperparameters which can be fixed by cross-validations. In experiment section, we will show our method is surprisingly robust to the λ, while the p should be carefully chosen. The hyperparameter λ is introduced to trade off between terms of data-fitting error and reconstruction error. When λ becomes large, the optimal subspace approximates the PCA subspace. Thus, we name our algorithm as PCA-based least square (PCA-LS).

3.2 PCA-Based Least Square Algorithm

We employ the traditional projection pursuit procedure [19,20] to incrementally construct the optimal subspace W and decision function g in problem (2). More specifically, we use an iterative procedure to minimize the objective function. In each iteration, based on the current model, we select one projection direction to add into the subspace and choose the coefficient in g for the selected direction such that the objective function has a maximum reduction.

one iteration in projection pursuit

Suppose that, at t^{th} iteration, we have $W = span\{w_1, \ldots, w_{t-1} | w_i \perp w_j, i \neq j\}$ and $g(v) = \sum_{j=1}^{t-1} \alpha_j v_j$. Then the residual r_i of decision response y_i is $r_i = y_i - \sum_{j=1}^{t-1} \alpha_j x_i^T w_j$, and the residual R_i of data point x_i is $R_i = x_i - x_i^W = x_i - \sum_{j=1}^{t-1} \beta_i^j w_j$. Note that R_i is orthogonal to the subspace W. Our goal in the t^{th} iteration is to optimize the following problem:

$$\begin{aligned}
\min_{\alpha,\beta,w} I(\alpha,\beta,w) &= \sum_{i=1}^{l}(y_i - \sum_{j=1}^{t-1} \alpha_j x_i^T w_j - \alpha x_i^T w)^2 + \sum_{i=1}^{n} \|x_i - \sum_{j=1}^{t-1} \beta_i^j w_j - \beta_i w\|^2 \\
&= \sum_{i=1}^{l}(r_i - \alpha x_i^T w)^2 + \lambda \sum_{i=1}^{n} \|R_i - \beta_i w\|^2 \\
&= \|r - \alpha {X^L}^T w\|^2 + \lambda \sum_{i=1}^{n} \|R_i - \beta_i w\|^2, \\
s.t. \quad & w \perp w_j \quad \forall j = 1, \ldots, t-1, \qquad (3)
\end{aligned}$$

where X^L is the first l columns of X, α is a scalar, $\beta = [\beta_1, \ldots, \beta_n]^T$ and $r = [r_1, \ldots, r_l]^T$. After w is solved from problem (3), we denote it by w_t and add it to W. In this way, we finish one iteration of projection pursuit.

The problem (3) is difficult to optimize due to the high order of variables and the orthogonal constraints. To eliminate the constraints, we limit the searching scope of direction to be the subspace spanned by the residuals, which means $w = \sum_{i=1}^{n} \eta_i R_i = R\eta$ in which $R = [R_1, \ldots, R_n]$ and $\eta = [\eta_1, \ldots, \eta_n]^T$. As $R_i \perp W \quad \forall i = 1, \ldots, n$, the orthogonal constraints are automatically met. It thus results in the following unconstrained minimization problem:

$$H(\alpha,\beta,\eta) = \|r - \alpha {X^L}^T R\eta\|^2 + \lambda \sum_{i=1}^{n} \|R_i - \beta_i R\eta\|^2. \qquad (4)$$

Fortunately, it is guaranteed that the optimal w^* of problem (3) is indeed a linear combination of R_i:

Proposition 1: The minimum point of problem (3) can be represented as a linear combination of R_i.

Proof: We decompose $\mathbb{R}^d$ as $\mathbb{R}^d = X^{\parallel} \oplus X^{\perp}$, where $X^{\parallel}$ is the subspace spanned by $x_i, 1 \leq i \leq n$ and $X^{\perp}$ is the orthogonal complement space of $X^{\parallel}$. By construction, $X^{\parallel}$ can be further decomposed as $X^{\parallel} = W \oplus R$, where W is the subspace spanned by $w_j, 1 \leq j \leq t-1$, and R is the subspace spanned by $R_i, 1 \leq i \leq n$. As the optimal solution w^* of (3) should be perpendicular to the subspace W, thus $w^* \in X^{\perp} \oplus R$. Assume $w^* = w^{\perp} + w^R$, where $w^{\perp} \in X^{\perp}$ and $w^R \in R$. So,

$$\begin{aligned} I(\alpha, \beta, w^*) &= \|r - \alpha X^{L^T} w^*\|^2 + \lambda \sum_{i=1}^{n} \|R_i - \beta_i w^*\|^2, \\ &= \|r - \alpha X^{L^T} (w^{\perp} + w^R)\|^2 + \lambda \sum_{i=1}^{n} \|R_i - \beta_i (w^{\perp} + w^R)\|^2, \\ &= \|r - \alpha X^{L^T} w^R\|^2 + \lambda \sum_{i=1}^{n} (\|R_i - \beta_i w^R\|^2 + \beta_i^2 \|w^{\perp}\|^2). \end{aligned} \tag{5}$$

The third equation follows from the fact that $X^{L^T} w^{\perp} = 0$ and $R_i^T w^{\perp} = 0$. Since $I(\alpha, \beta, w^*) \geq I(\alpha, \beta, w^R)$ and w^* minimize $I(\alpha, \beta, w)$, we have $w^* = w^R$. Thus, w^* is in the subspace R, and can be represented as a linear combination of $R_i, 1 \leq i \leq n$. ■

To minimize the objective function (4), the iterative coordinate decent method is used. Briefly speaking, in each step, we optimize α, β for fixed η , and then optimize η for fixed α, β.

For a fixed η, the optimal α and β can be obtained by setting the partial derivatives $\frac{\partial H(\alpha,\beta,\eta)}{\partial \alpha}$, $\frac{\partial H(\alpha,\beta,\eta)}{\partial \beta}$ to zeros, and are given by:

$$\alpha = \frac{\langle r, X^{L^T} R\eta \rangle}{\langle X^{L^T} R\eta, X^{L^T} R\eta \rangle}, \qquad \beta_i = \frac{\langle R_i, R\eta \rangle}{\langle R\eta, R\eta \rangle}. \tag{6}$$

For fixed α and β, gradient decent is used to update η. The partial derivative of $H(\alpha, \beta, \eta)$ with respect to η is given by

$$\frac{\partial H(\alpha, \beta, \eta)}{\partial \eta} = -2\alpha R^T X^L r + 2\alpha^2 R^T X^L X^{L^T} R\eta + 2\lambda R^T R((\sum_{i=1}^{n} \beta_i^2)\eta - \beta). \tag{7}$$

After the iterative coordinate decent method converges, we get the optimal solution $\alpha^*, \beta^*, \eta^*$ for the problem (4). The new projection direction $w_t = R\eta^*$ is then added into $\{w_1, \ldots, w_{t-1}\}$ to form the new basis of subspace W. The residual of the response and the residual of inputs are updated by $r \leftarrow r - \alpha^* X^{L^T} R\eta^*$ and $R_i \leftarrow R_i - \beta_i^* R\eta^*$. Note that the new residual R_i preserves the property that it is orthogonal to the new subspace $W = span\{w_1, \ldots, w_t\}$. This fact follows the observation that the method we use to update the residual of input data is exactly the Gram-Schmidt orthogonalization.

After p times greedy search, we get the p dimensional subspace W and a decision function $g(v) = \sum_{t=1}^{p} \alpha_t v_t$ defined on W. If only classification is concerned, these two things can be combined to get the final decision function defined on the input space $f(x) = \sum_{t=1}^{p} \alpha_t x^T w_t = x^T W \overline{\alpha}$.

The whole procedure is summarized in Algorithm 1.

Algorithm 1. PCA-based Least Square (PCA-LS)

```
Init: $r = [r_1, \dots, r_l]^T$;   $R_i = x_i \quad i = 1, \dots, n$
for $t = 1$ to $p$ do
  repeat
    1. Compute $\alpha, \beta$ using (6)
    2. Compute $\frac{\partial H(\alpha,\beta,\eta)}{\partial \eta}$ using (7)
    3. $\eta = \eta - StepSize * \frac{\partial H(\alpha,\beta,\eta)}{\partial \eta}$
  until $\eta$ is convergent
  $w_t = R\eta$
  $\alpha_t = \alpha$
  $r = r - \alpha_t X^L w_t$
  $R_i = R_i - \beta_i w_t \quad i = 1, \dots, n$
end for
Output:
$f(x) = \sum_{t=1}^{p} \alpha_t x^T w_t = x^T W \overline{\alpha}$
$\overline{\alpha} = [\alpha_1, \dots, \alpha_p]^T$
```

3.3 Kernel PCA-Based Least Square

When the data set has highly nonlinear structure, the PCA-based least square may fail. One common technique to tackle the nonlinear problem in machine learning is the kernel trick, which can be briefly described as follows: with a feature mapping $\phi : x \rightarrow \phi(x)$, the input data is mapped to a feature space. For linear learning in this feature space, the inner product of two mapped data is defined as the kernel function: $k(x, y) = \langle \phi(x), \phi(y) \rangle$, and the matrix K with $(K)_{ij} = k(x_i, x_j)$ is the Gram matrix.

At the t^{th} iteration, suppose that the residual R_i of $\phi(x_i)$ can be expressed as $R_i = \sum_{j=1}^{n} M_i^j \phi(x_j) = \phi(X) M_i$, where $\phi(X) = [\phi(x_1), \dots, \phi(x_n)]$ and $M_i = [M_i^1, \dots, M_i^n]^T$. Thus, $R = [R_1, \dots, R_n] = \phi(X)[M_1, \dots, M_n] = \phi(X)M$, where M is a $n \times n$ matrix. In parallel with the linear case, we constrain the projection direction w to be a linear combination of residuals: $w = \sum_{i=1}^{n} \eta_i R_i = R\eta = \phi(X)M\eta$. Now we get the objective function $H(\alpha, \beta, \eta)$ of the kernel method similar to the linear case:

$$H(\alpha, \beta, \eta) = \|r - \alpha \phi(X^L)^T R\eta\|^2 + \lambda \sum_{i=1}^{n} \|R_i - \beta_i R\eta\|^2$$

$$= \|r - \alpha\phi(X^L)^T\phi(X)M\eta\|^2 + \lambda\sum_{i=1}^{n}\|R_i - \beta_i R\eta\|^2$$
$$= \|r - \alpha K^{L^T} M\eta\|^2 + \lambda\sum_{i=1}^{n}\|R_i - \beta_i R\eta\|^2, \tag{8}$$

where K^L is the first l columns of the Gram matrix K.

As before, we employ the iterative coordinate decent method to minimize the objective function (8). In each step, α, β are given by

$$\alpha = \frac{r^T K^{L^T} M\eta}{\eta^T M^T K^L K^{L^T} M\eta}, \qquad \beta_i = \frac{M_i K M\eta}{\eta^T M^T K M\eta}, \tag{9}$$

and the partial derivative of $H(\alpha, \beta, \eta)$ with respect to η is given by

$$\frac{\partial H(\alpha,\beta,\eta)}{\partial\eta} = -2\alpha M^T K^L r + 2\alpha^2 M^T K^L K^{L^T} M\eta + 2\lambda M^T K M((\sum_{i=1}^{n}\beta_i^2)\eta - \beta). \tag{10}$$

After the iterative coordinate decent method converges, we get the optimal solution $\alpha^*, \beta^*, \eta^*$ for the problem (8). Then, the direction $w_t = \phi(X)M\eta^*$ is added to $\{w_1, \ldots, w_{t-1}\}$ to form the new basis of subspace W. The residual of response and residual of inputs are updated by: $r \leftarrow r - \alpha^*\phi(X^L)^T R\eta^* = r - \alpha^* K^{L^T}\eta^*$ and $M_i \leftarrow M_i - \beta_i^* M\eta^*$. Again the new residual R_i is orthogonal to the new subspace $W = span\{w_1, \ldots, w_t\}$.

The whole process is summarized in Algorithm 2. Different from the linear case, we can not express the projection direction w_t explicitly. Instead, the coefficient vector s_t of w_t's linear representation by $\{\phi(x_1), \ldots, \phi(x_n)\}$ is stored.

3.4 Computational Complexity

As discussed above, large-scale problem is extremely important for semi-supervised learning. The linear PCA-LS algorithm consists of p times greedy search iterations, where p is the dimensionality of the subspace W. The complexity of every iteration is dominated by computing the gradient of η which scales as $O(nd)$ where d is the dimensionality of the input space. Thus, the computational complexity of the linear PCA-LS is $O(pnd)$. By a similar analysis, the computational complexity of the kernel PCA-LS algorithm scales as $O(pn^2)$.

4 Experiments

In this section, we first conduct experiments on synthetic data sets to show the ability of subspace regularization methods in handling overlapping data and manifold data. Then comparison experiments are given on several real data sets. Finally, we analyze the robustness of our methods to hyperparameters.

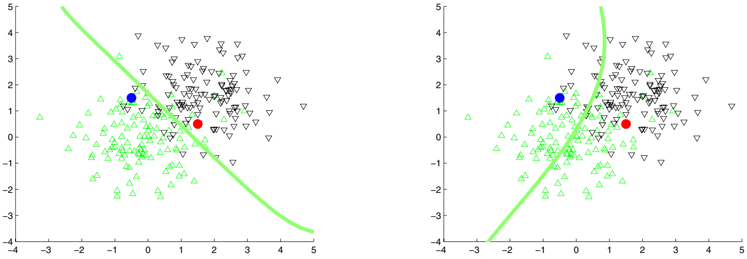
Algorithm 2. Kernel PCA-based Least Square (Kernel PCA-LS)

```
**Init:** $r = [r_1, \ldots, r_l]^T$; $M = I$
**for** $t = 1$ **to** $p$ **do**
  **repeat**
    1. Compute $\alpha, \beta$ using (9)
    2. Compute $\frac{\partial H(\alpha,\beta,\eta)}{\partial \eta}$ using (10)
    3. $\eta = \eta - StepSize * \frac{\partial H(\alpha,\beta,\eta)}{\partial \eta}$
  **until** $\eta$ is convergent
  $s_t = M\eta$
  $\alpha_t = \alpha$
  $r = r - \alpha_t K^{L^T} s_t$
  $M_i = M_i - \beta_i s_t \quad i = 1, \ldots, n$
**end for**
**Output:**
$f(x) = k_n(x)^T S\overline{\alpha}$
$k_n(x) = [k(x, x_1), \ldots, k(x, x_n)]^T$
$S = [s_1, \ldots, s_p]$
$\overline{\alpha} = [\alpha_1, \ldots, \alpha_p]^T$
```

4.1 Overlapping Data

When data of different classes overlap heavily, the optimal decision boundary or Bayesian decision boundary may cross the overlapping area which is a high density region. In this case, the smoothness-based methods tend to fail as they prefer a decision boundary located in low density area [3,8]. With following experiments, we demonstrate that subspace regularization methods can effectively avoid this problem.

Two dimension case. 200 data points were drawn from each of two unit-variance Gaussians, the centers of which are $(0, 0)$ and $(1.5, 1.5)$. Only one point



(a) (b)

Fig. 1. Two Gaussians Data Set: Decision boundary using RBF kernel for KPCA-LS (a) and KLapRLS (b)

was labeled in each class and all the rest points were treated as unlabeled. The data set and the best decision boundary across a range of hyperparameters of kernel PCA-LS (KPCA-LS) and kernel LapRLS (KLapRLS) [6] are shown in Figure 1. Just as PCA, KPCA-LS finds the first principal component direction as the subspace to project on. In Figure 1 (a), this direction is parallel to the line passing through the centers of two Gaussians. As a result, the decision boundary is a line that is perpendicular to the selected subspace and can correctly classify the labeled data. However, for KLapRLS, the smoothness regularization term makes the decision boundary avoid the overlapping area as crossing such a high density region would lead to a big penalty. This makes the KLapRLS fail to find a satisfactory solution.

High dimension case. G241c and G241d are commonly-used data sets in semi-supervised learning which are constructed by Chapelle et al. [1]. Each of them is composed of 1500 points with dimension of 241. For G241c, two classes data come from two unit-variance Gaussians respectively, and centers of the two Gaussians have a distance of 2.5. For G241d, the data of the first class come from the two unit-variance Gaussians, the centers of which have a distance of 6; and the data of second class come from another two Gaussians which are fixed by moving each of the former centers a distance of 2.5. By the construction of G241c and G241d, we see that there exists overlapping between different class.

We compare subspace regularization methods with 3 smoothness-based methods, including Gaussian Random Field (GRF) [5], Learning with Local and Global Consistency (LLGC) [2] and manifold regularization (linear LapRLS and kernel LapRLS) [6]. 50 points are randomly sampled as labeled data, and the rest are left as unlabeled data. Data set is split 10 times, and the reported results are averaged classification accuracy over these 10 splits. 5-fold cross-validation is used to select hyperparameters, and the detailed setting for every method is introduced in following section. Experiments results is summarized in Table 1.

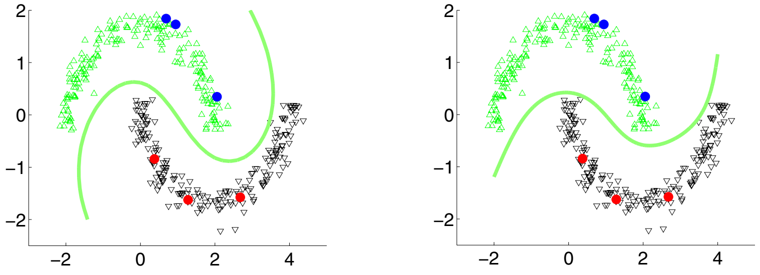
From the results, we can conclude that, when data of different classes overlap heavily, the subspace regularization methods outperform smoothness-based methods dramatically.

4.2 Manifold Data

This experiment was conducted on the two moons data set which was designed to satisfy the smoothness assumption and have a strong geometric structure. 200 data points were drawn from each of two moons, and three points drawn from

Table 1. Averaged classification accuracy (%) for 50 labeled data

	GRF	LLGC	LapRLS	KLapRLS	PCA-LS	KPCA-LS
G241c	48.28	48.28	71.66	68.36	**84.62**	83.59
G241d	65.62	63.53	67.90	66.28	**83.92**	74.01



(a)

(b)

Fig. 2. Two Moons Data Set: Decision boundary using RBF kernel for KPCA-LS (a) and KLapRLS (b)

each class were labeled. The data set and the best decision boundary across a range of hyperparameters of kernel PCA-LS and kernel LapRLS are shown in Figure 2. It is well-known that the kernel LapRLS algorithm can work perfectly well on such problems. As can be seen from Figure 2, the kernel PCA-LS also learns a satisfactory decision function that correctly classify all the data points.

4.3 Real Data Sets

To evaluate our methods, We perform experiments on seven data sets of binary classification problem that come from the benchmark in Chapelle's book [1] and UCI machine learning repository. These data set originate from areas like image, text, biometrics etc., with size from 277 to 83,679 and dimension from 9 to 11,960. The characteristics of the data sets are shown in Table 2.

We compare subspace regularization methods with 3 smoothness-based semi-supervised learning methods, including GRF [5], LLGC [2], manifold regularization (linear LapRLS and kernel LapRLS) [6]. We also use both labeled and unlabeled data to perform dimension reduction with PCA, and then use regularized least square (RLS) to learn classifier on labeled data. For convenience, we name this baseline algorithm as PCA+RLS. We details the experimental setting for each method in the following:

Table 2. Description of the data sets

	Breast-cancer	BCI	USPS	Text	Image	Waveform	SecStr
size	277	400	1500	1500	2310	5000	83,679
dimension	9	117	241	11,960	18	21	315

Table 3. Averaged classification accuracy (%)

	PCA+RLS	GRF	LLGC	LapRLS	KLapRLS	PCA-LS	KPCA-LS
Breast-cancer	67.84	70.83	70.22	67.36	70.53	**73.44**	71.98
BCI	57.77	50.34	50.00	**62.29**	61.69	**62.29**	58.94
USPS	86.18	89.81	**94.5**	84.61	87.48	84.10	86.18
Text	**73.25**	52.68	62.63	71.84	68.62	72.25	71.32
Image	78.00	49.53	73.17	76.90	**83.60**	76.36	80.13
Waveform	78.99	70.84	79.69	78.34	-	85.33	**86.18**
SecStr	60.26	-	-	60.41	-	**61.28**	-

- PCA-LS: the dimension of subspace is chosen from $\{2; 2^2; 2^3; 2^4; 2^5\}$, and the regularization factor λ is simply fixed to 1.
- KPCA-LS: the dimension of subspace is chosen from $\{2; 2^2; 2^3; 2^4; 2^5\}$, the regularization factor λ is simply fixed to 100. RBF kernel is selected as kernel function, and the σ_k is chosen from $\sigma_0 * \{2; 1; 2^{-1}; 2^{-2}; 2^{-3}; 2^{-4}\}$ where σ_0 is the averaged distance between every two points in data set.
- GRF and LLGC methods: we use RBF kernel as similarity function to construct the graph, and the hyperparameter σ_g is chosen from $\sigma_0 * \{2; 1; 2^{-1}; 2^{-2}; 2^{-3}; 2^{-4}\}$ where σ_0 is defined as above.
- LapRLS: γ_A and γ_I are chosen from $\{10^4; 10^2; 1; 10^{-2}; 10^{-4}\}$. Linear kernel is used as kernel function. For all data sets except SecStr, RBF kernel is used as similarity function to construct the graph, and the hyperparameter σ_g is chosen from $\sigma_0 * \{2; 1; 2^{-1}; 2^{-2}; 2^{-3}; 2^{-4}\}$ where σ_0 is defined as above. For SecStr, the graph entry $G_{ij} = 1$ if x_i is among x_j's k nearest neighbors, else $G_{ij} = 0$. k is fixed to 5 which is the value used in [1].
- KLapRLS: γ_A, γ_I and σ_g is selected as in linear LapRLS. RBF kernel is used as kernel function, and the σ_k is chosen from $\sigma_0 * \{2; 1; 2^{-1}; 2^{-2}; 2^{-3}; 2^{-4}\}$.
- PCA+RLS: the dimension of subspace is chosen from $\{2; 2^2; 2^3; 2^4; 2^5\}$, and the regularization factor λ is chosen from $\{10^4; 10^2; 1; 10^{-2}; 10^{-4}\}$.

As some graph-based methods like GRF and LLGC can not directly predict the out-of-sample points, we adopt the transductive setting in the experiments which means all the data points are available before training. But we emphasize that the subspace regularization method is inductive.

For all data sets except SecStr, 50 points are randomly sampled as labeled data, and the rest are left unlabeled. We split each data set 10 times, and report the averaged classification accuracy over these 10 splits. 5-fold cross-validation is used to select hyperparameters. For SecStr, 500 points are randomly sampled as labeled data in which 400 points are used for training and 100 points are used to select the hyperparameters. The reported results are averaged accuracy on 10 splits.

Table 3 summarizes the experiments results. There are blank entries in Table 3 as some algorithms require too many memory, or the results can not be achieved within 3 days.

For all data sets except image (Image and USPS), subspace regularization method works as well as or better than smoothness-based methods. It demonstrates that subspace regularization is a powerful semi-supervised learning method. We will analyze this result in more detail in the next section.

For SecStr data set, it costs PCA-LS less than 5 minutes to learn a decision function and a subspace of dimension $p = 8$ which is the dimensionality selected by most splits. However, for graph-based methods, only to construct a neighborhood graph will take more than 2 hours without saying to inverse a matrix of same size with neighborhood graph. All experiments are performed on a PC with 3.0GHz CPU, 2GB memory, using Matlab.

4.4 Robustness Analysis

Recall that in all of the above experiments the regularization factor λ for subspace regularization methods is set to a fixed number, and here we will examine this robustness of our methods to hyperparameters in more details.

To evaluate the PCA-LS's robustness to the regularization factor λ, we fix the dimension p of subspace to 16 and report the classification accuracy for $\lambda = \{10^{-4}, 10^{-2}, 10^{0}, 10^{2}, 10^{4}\}$. For SecStr, 500 points are randomly labeled, and for other data sets 50 data are randomly labeled. The result is averaged over 10 splits which is shown in the Figure 3. We can see that PCA-LS is rather robust to the variation of λ, so no careful tuning of λ is needed.

To evaluate the PCA-LS's robustness to the dimension p of subspace, we fix the λ to 1 and report the classification accuracy for $p = \{1, 2, 2^2, 2^3, 2^4, 2^5, 2^6\}$. For SecStr, 500 points are randomly labeled, and for other data sets 50 data are randomly labeled. The result shown in the Figure 4 is averaged over 10 splits.

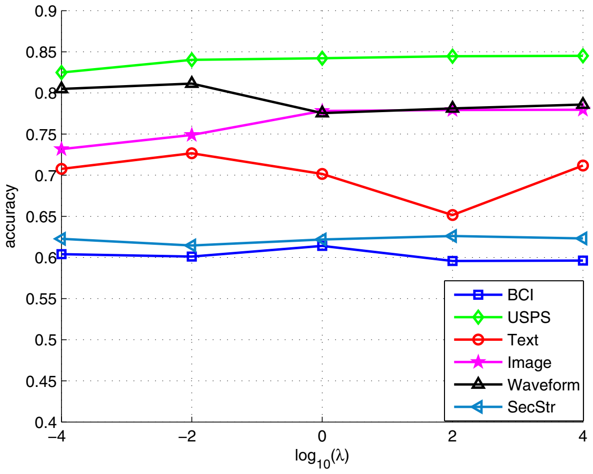


Fig. 3. PCA-LS's classification accuracy with different λ ($p = 16$)

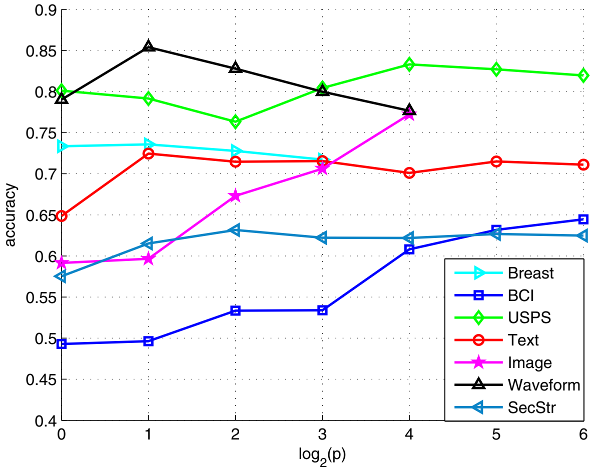


Fig. 4. PCA-LS's classification accuracy with different p $(\lambda = 1)$

Unlike λ, the dimension p of the subspace is very important to the success of subspace regularization methods, and needs to be chosen carefully. According to the performance of PCA-LS, these seven data sets can be clearly divided into two categories. For Breast-cancer, waveform, Text and SecStr, the accuracy peaks at a very small p and then decrease gradually, which means most of the information valuable for classification is actually embedding in a low dimensional subspace. For Image, USPS and BCI, however, the accuracy increases with the dimension of subspace monotonically which means the discriminative information is highly nonlinear and can not be captured by a low dimensional subspace. It is very interesting to see that it is exactly on the first class of data sets that the subspace regularization methods beat all the smoothness-based methods, while on the second class of data sets smoothness-based methods work better. Thus the conclusion is that, for very limited labeled data finding a general semi-supervised learning method that works well for most problem may be extremely hard, and we need to choose method carefully according to the characteristics of problem at hands. For manifold or highly nonlinear data set, smoothness-based methods are good choices. For those linear or approximately linear problems, subspace regularization methods are more effective.

5 Conclusions

This paper has presented subspace regularization, a new method for semi-supervised learning. The motivation of our work is to find a low-dimensional representation of data to avoid the curse of dimensionality and reduce the complexity of the problem. Specifically, we hope to find a subspace and a decision function defined on this subspace such that the projection of labeled data onto this subspace can be easily separated and meanwhile the data information does not loss too much by projection. By specifying the regularization term as reconstruction error as PCA, we propose the PCA-based least square algorithm.

Unlike most of the semi-supervised learning methods which are based on the smoothness assumption, subspace regularization utilizes the classic low dimensional embedding assumption. Compared with previous works, our methods have two remarkable advantages. First, under the situation that data from different classes overlap heavily, our methods can still work, while smoothness-based methods may fail. Second, our method has low computational complexity. For linear PCA-LS, the computational complexity is linear in the number of data points and dimension of the input space. This favorable property enables our method to be applied to large-scale applications which have been demonstrated in the experiment section.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant nos. 60825301 and 60723005. The authors thank the anonymous reviewers for their valuable comments.

References

1. Chapelle, O., Schölkopf, B., Zien, A., NetLibrary, I.: Semi-supervised learning. MIT Press, Cambridge (2006)
2. Zhou, D., Bousquet, O., Lal, T., Weston, J., Scholkopf, B.: Learning with local and global consistency. In: Advances in Neural Information Processing Systems 16, pp. 321–328 (2004)
3. Mann, G., McCallum, A.: Simple, robust, scalable semi-supervised learning via expectation regularization. In: Proceedings of the International Conference on Machine Learning, pp. 593–600 (2007)
4. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02, Computer Science, University of Wisconsin-Madison (2002)
5. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using Gaussian fields and Harmonic functions. In: Proceedings of the International Conference on Machine Learning, pp. 912–919 (2003)
6. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. The Journal of Machine Learning Research 7, 2399–2434 (2006)
7. Joachims, T.: Transductive inference for text classification using support vector machines. In: Proceedings of the International Conference on Machine Learning, pp. 200–209 (1999)
8. Zhu, X.: Semi-supervised learning literature survey. Technical report, Computer Science, University of Wisconsin-Madison (2005)
9. Burges, C.: Geometric methods for feature extraction and dimensional reduction: A guided tour. In: The Data Mining and Knowledge Discovery Handbook, pp. 59–92 (2005)
10. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1995)

11. Jolliffe, I.: Principal component analysis. Springer, New York (2002)
12. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. Machine learning 39(2), 103–134 (2000)
13. Joachims, T.: Transductive learning via spectral graph partitioning. In: Proceedings of the International Conference on Machine Learning, pp. 290–297 (2003)
14. Belkin, M., Niyogi, P.: Semi-supervised learning on Riemannian manifolds. Machine Learning 56(1), 209–239 (2004)
15. Lawrence, N.D., Jordan, M.I.: Semi-supervised learning via Gaussian processes. In: Saul, L.K., Weiss, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems 17, pp. 753–760 (2005)
16. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: Advances in Neural Information Processing Systems 17, pp. 529–536 (2005)
17. Szummer, M., Jaakkola, T.: Information regularization with partially labeled data. In: Becker, S., Thrun, S., Obermayer, K. (eds.) Advances in Neural Information Processing Systems 15 (2003)
18. Chapelle, O., Zien, A.: Semi-supervised classification by low density separation. In: Cowell, R.G., Ghahramani, Z. (eds.) Society for Artificial Intelligence and Statistics, pp. 57–64 (2005)
19. Friedman, J., Stuetzle, W.: Projection pursuit regression. Journal of the American Statistical Association 76(376), 817–823 (1981)
20. Huber, P.: Projection pursuit. The Annals of Statistics 13(2), 435–475 (1985)