# Semantic Concept Classification by Joint Semi-supervised Learning of Feature Subspaces and Support Vector Machines

Wei Jiang[1], Shih-Fu Chang[1], Tony Jebara[1], and Alexander C. Loui[2]

[1] Columbia University, New York, NY 10027, USA
[2] Eastman Kodak Company, Rochester, NY 14650, USA

**Abstract.** The scarcity of labeled training data relative to the high-dimensionality multi-modal features is one of the major obstacles for semantic concept classification of images and videos. Semi-supervised learning leverages the large amount of unlabeled data in developing effective classifiers. Feature subspace learning finds optimal feature subspaces for representing data and helping classification. In this paper, we present a novel algorithm, Locality Preserving Semi-supervised Support Vector Machines (LPSSVM), to jointly learn an optimal feature subspace as well as a large margin SVM classifier. Over both labeled and unlabeled data, an optimal feature subspace is learned that can maintain the smoothness of local neighborhoods as well as being discriminative for classification. Simultaneously, an SVM classifier is optimized in the learned feature subspace to have large margin. The resulting classifier can be readily used to handle unseen test data. Additionally, we show that the LPSSVM algorithm can be used in a Reproducing Kernel Hilbert Space for nonlinear classification. We extensively evaluate the proposed algorithm over four types of data sets: a toy problem, two UCI data sets, the Caltech 101 data set for image classification, and the challenging Kodak's consumer video data set for semantic concept detection. Promising results are obtained which clearly confirm the effectiveness of the proposed method.

## 1 Introduction

Consider one of the central issues in semantic concept classification of images and videos: the amount of available unlabeled test data is large and growing, but the amount of labeled training data remains relatively small. Furthermore, the dimensionality of the low-level feature space is generally very high, the desired classifiers are complex and, thus, small sample learning problems emerge.

There are two primary techniques for tackling above issues. *Semi-supervised learning* is a method to incorporate knowledge about unlabeled test data into the training process so that a better classifier can be designed for classifying test data [1], [2], [3], [4], [5]. *Feature subspace learning*, on the other hand, tries to learn a suitable feature subspace for capturing the underlying data manifold over which distinct classes become more separable [6], [7], [8], [9].

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part IV, LNCS 5305, pp. 270–283, 2008.

One emerging branch of semi-supervised learning methods is graph-based techniques [2], [4]. Within a graph, the nodes are labeled and unlabeled samples, and weighted edges reflect the feature similarity of sample pairs. Under the assumption of label smoothness on the graph, a discriminative function $f$ is often estimated to satisfy two conditions: the loss condition – it should be close to given labels $y_L$ on the labeled nodes; and the regularization condition – it should be smooth on the whole graph, i.e., close points in the feature space should have similar discriminative functions. Among these graph-based methods, *Laplacian Support Vector Machines* (*LapSVM*) and *Laplacian Regularized Least Squares* (*LapRLS*) are considered state-of-the-art for many tasks [10]. They enjoy both high classification accuracy and extensibility to unseen out-of-sample data.

Feature subspace learning has been shown effective for reducing data noise and improving classification accuracy [6], [7], [8], [9]. Finding a good feature subspace can also improve semi-supervised learning performance. As in classification, feature subspaces can be found by supervised methods (e.g., LDA [8]), unsupervised methods (e.g., graph-based manifold embedding algorithms [6], [9]), or semi-supervised methods (e.g., generalizations of graph-based embedding by using the ground-truth labels to help the graph construction process [7]).

In this paper, we address both issues of feature subspace learning and semi-supervised classification. We pursue a new way of feature subspace and classifier learning in the semi-supervised setting. A novel algorithm, *Locality Preserving Semi-supervised SVM* (*LPSSVM*), is proposed to jointly learn an optimal feature subspace as well as a large margin SVM classifier in a semi-supervised manner. A joint cost function is optimized to find a smooth and discriminative feature subspace as well as an SVM classifier in the learned feature subspace. Thus, the local neighborhoods relationships of both labeled and unlabeled data can be maintained while the discriminative property of labeled data is exploited. The following highlight some aspects of the proposed algorithm:

1. The target of LPSSVM is both feature subspace learning and semi-supervised classification. A feature subspace is jointly optimized with an SVM classifier so that in the learned feature subspace the labeled data can be better classified with the optimal margin, and the locality property revealed by both labeled and unlabeled data can be preserved.
2. LPSSVM can be readily extended to classify novel unseen test examples. Similar to LapSVM and LapRLS and other out-of-sample extension methods [5], [10], this extends the algorithm's flexibility in real applications, in contrast with many traditional graph-based semi-supervised approaches [4].
3. LPSSVM can be learned in the original feature space or in a Reproducing Kernel Hilbert Space (RKHS). In other words, a kernel-based LPSSVM is formulated which permits the method to handle real applications where nonlinear classification is often needed.

To evaluate the proposed LPSSVM algorithm, extensive experiments are carried out over four different types of data sets: a toy data set, two UCI data sets [11], the Caltech 101 image data set for image classification [12], and the large scale Kodak's consumer video data set [13] from real users for video concept detection. We compare our algorithm with several state of the arts, including the standard SVM [3], semi-supervised LapSVM and LapRLS [10], and the naive approach of first learning a feature subspace (unsupervised) and then solving an SVM (supervised) in the learned feature subspace. Experimental results demonstrate the effectiveness of our LPSSVM algorithm.

## 2 Related Work

Assume we have a set of data points $X=[\mathbf{x}_1,\ldots,\mathbf{x}_n]$, where $\mathbf{x}_i$ is represented by a $d$-dimensional feature vector, i.e., $\mathbf{x}_i \in \mathbb{R}^d$. $X$ is partitioned into labeled subset $X_L$ (with $n_L$ data points) and unlabeled subset $X_U$ (with $n_U$ data points), $X=[X_L, X_U]$. $y_i$ is the class label of $\mathbf{x}_i$, e.g., $y_i \in \{-1, +1\}$ for binary classification.

### 2.1 Supervised SVM Classifier

The SVM classifier [3] has been a popular approach to learn a classifier based on the labeled subset $X_L$ for classifying the unlabeled set $X_U$ and new unseen test samples. The primary goal of an SVM is to find an optimal separating hyperplane that gives a low generalization error while separating the positive and negative training samples. Given a data vector $\mathbf{x}$, SVMs determine the corresponding label by the sign of a linear decision function $f(\mathbf{x})=\mathbf{w}^T\mathbf{x}+b$. For learning nonlinear classification boundaries, a kernel mapping $\phi$ is introduced to project data vector $\mathbf{x}$ into a high dimensional feature space as $\phi(\mathbf{x})$, and the corresponding class label is given by the sign of $f(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x})+b$. In SVMs, this optimal hyperplane is determined by giving the largest margin of separation between different classes, i.e. by solving the following problem:

$$\min_{\mathbf{w},b,\epsilon} Q_d = \min_{\mathbf{w},b,\epsilon}\left\{\frac{1}{2}||\mathbf{w}||_2^2 + C\sum_{i=1}^{n_L}\epsilon_i\right\}, s.t.\ y_i(\mathbf{w}^T\phi(\mathbf{x}_i)+b)\geq 1-\epsilon_i, \epsilon_i\geq 0, \forall\ \mathbf{x}_i \in X_L\ . \quad (1)$$

where $\epsilon=\epsilon_1,\ldots,\epsilon_{n_L}$ are the slack variables assigned to training samples, and $C$ controls the scale of the empirical error loss the classifier can tolerate.

### 2.2 Graph Regularization

To exploit the unlabeled data, the idea of graph Laplacian [6] has been shown promising for both subspace learning and classification. We briefly review the ideas and formulations in the next two subsections. Given the set of data points $X$, a weighted undirected graph $G=(V,E,W)$ can be used to characterize the pairwise similarities among data points, where $V$ is the vertices set and each node $v_i$ corresponds to a data point $\mathbf{x}_i$; $E$ is the set of edges; $W$ is the set of weights measuring the strength of the pairwise similarity.

**Regularization for feature subspace learning.** In feature subspace learning, the objective of graph Laplacian [6] is to embed original data graph into an $m$-dimensional Euclidean subspace which preserves the locality property of original data. After embedding, connected points in original $G$ should stay close. Let $\hat{X}$ be the $m{\times}n$ dimensional embedding, $\hat{X}{=}[\hat{\mathbf{x}}_1,\ldots,\hat{\mathbf{x}}_n]$, the cost function is:

$$\min_{\hat{X}}\left\{\sum_{i,j=1}^{n}||\hat{\mathbf{x}}_i-\hat{\mathbf{x}}_j||_2^2W_{ij}\right\}, s.t.\hat{X}D\hat{X}^T{=}I \Rightarrow \min_{\hat{X}}\left\{\mathrm{tr}(\hat{X}L\hat{X}^T)\right\}, s.t.\hat{X}D\hat{X}^T{=}I. \quad (2)$$

where $L$ is the Laplacian matrix and $L{=}D{-}W$, $D$ is the diagonal weight matrix whose entries are defined as $D_{ii}=\sum_j W_{ij}$. The condition $\hat{X}D\hat{X}^T{=}I$ removes an arbitrary scaling factor in the embedding [6]. The optimal embedding can be obtained as the matrix of eigenvectors corresponding to the lowest eigenvalues of the generalized eigenvalue problem: $L\hat{\mathbf{x}}{=}\lambda D\hat{\mathbf{x}}$. One major issue of this graph embedding approach is that when a novel unseen sample is added, it is hard to locate the new sample in the embedding graph. To solve this problem, the *Locality Preserving Projection* (*LPP*) is proposed [9] which tries to find a linear projection matrix $\mathbf{a}$ that maps data points $\mathbf{x}_i$ to $\mathbf{a}^T\mathbf{x}_i$, so that $\mathbf{a}^T\mathbf{x}_i$ can best approximate graph embedding $\hat{\mathbf{x}}_i$. Similar to Eq(2), the cost function of LPP is:

$$\min_{\mathbf{a}} Q_s = \min_{\mathbf{a}}\left\{\mathrm{tr}(\mathbf{a}^TXLX^T\mathbf{a})\right\}, \quad s.t.\ \mathbf{a}^TXDX^T\mathbf{a} = I\ . \quad (3)$$

We can get the optimal projection as the matrix of eigenvectors corresponding to the lowest eigenvalues of generalized eigenvalue problem: $XLX^T\mathbf{a} = \lambda XDX^T\mathbf{a}$.

**Regularization for classification.** The idea of graph Laplacian has been used in semi-supervised classification, leading to the development of Laplacian SVM and Laplacian RLS [10]. The assumption is that if two points $\mathbf{x}_i, \mathbf{x}_j \in X$ are close to each other in the feature space, then they should have similar discriminative functions $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$. Specifically the following cost function is optimized:

$$\min_f \frac{1}{n_L}\sum\nolimits_{i=1}^{n_L} V(\mathbf{x}_i, y_i, f) + \gamma_A||f||_2^2 + \gamma_I\mathbf{f}^TL\mathbf{f}\ . \quad (4)$$

where $V(\mathbf{x}_i,y_i,f)$ is the loss function, e.g., the square loss $V(\mathbf{x}_i,y_i,f){=}(y_i{-}f(\mathbf{x}_i))^2$ for LapRLS and the hinge loss $V(\mathbf{x}_i,y_i,f) = \max(0, 1-y_if(\mathbf{x}_i))$ for LapSVM; $\mathbf{f}$ is the vector of discriminative functions over the entire data set $X$, i.e., $\mathbf{f} = [f(\mathbf{x}_1),\ldots,f(\mathbf{x}_{n_U+n_L})]^T$. Parameters $\gamma_A$ and $\gamma_I$ control the relative importance of the complexity of $f$ in the ambient space and the smoothness of $f$ according to the feature manifold, respectively.

### 2.3 Motivation

In this paper, we pursue a new semi-supervised approach for feature subspace discovery as well as classifier learning. We propose a novel algorithm, *Locality Preserving Semi-supervised SVM* (*LPSSVM*), aiming at joint learning of both an optimal feature subspace and a large margin SVM classifier in a semi-supervised manner. Specifically, the graph Laplacian regularization condition in Eq(3) is adopted to maintain the smoothness of the neighborhoods over both labeled and unlabeled data. At the same time, the discriminative constraint in Eq(1) is used to maximize the discriminative property of the learned feature subspace over the labeled data. Finally, through optimizing a joint cost function, the semi-supervised feature subspace learning and semi-supervised classifier learning can work together to generate a smooth and discriminative feature subspace as well as a large-margin SVM classifier.

In comparison, standard SVM does not consider the manifold structure presented in the unlabeled data and thus usually suffers from small sample learning problems. The subspace learning methods (e.g. LPP) lack the benefits of large margin discriminant models. Semi-supervised graph Laplacian approaches, though incorporating information from unlabeled data, do not exploit the advantage of feature subspace discovery. Therefore, the overarching motivation of our approach is to jointly explore the merit of feature subspace discovery and large-margin discrimination. We will show through four sets of experiments such approach indeed outperforms the alternative methods in many classification tasks, such as semantic concept detection in challenging image/video sets.

## 3 Locality Preserving Semi-supervised SVM

In this section we first introduce the linear version of the proposed LPSSVM technique then show it can be readily extended to a nonlinear kernel version.

### 3.1 LPSSVM

The smooth regularization term $Q_s$ in Eq(3) and discriminative cost function $Q_d$ in Eq(1) can be combined synergistically to generate the following cost function:

$$\min_{\mathbf{a},\mathbf{w},b,\epsilon} Q = \min_{\mathbf{a},\mathbf{w},b,\epsilon}\{Q_s + \gamma Q_d\} = \min_{\mathbf{a},\mathbf{w},b,\epsilon}\left\{\text{tr}(\mathbf{a}^T XLX^T\mathbf{a}) + \gamma[\frac{1}{2}||\mathbf{w}||_2^2 + C\sum\nolimits_{i=1}^{n_L}\epsilon_i]\right\} \quad (5)$$
$$s.t.\ \mathbf{a}^T XDX^T\mathbf{a} = I, \quad y_i(\mathbf{w}^T\mathbf{a}^T\mathbf{x}_i + b) \geq 1 - \epsilon_i,\ \epsilon_i \geq 0,\ \forall\ \mathbf{x}_i \in X_L\ .$$

Through optimizing Eq(5) we can obtain the optimal linear projection $\mathbf{a}$ and classifier $\mathbf{w}, b$ simultaneously. In the following, we develop an iterative algorithm to minimize over $\mathbf{a}$ and $\mathbf{w}, b, \epsilon$ which will monotonically reduce the cost $Q$ by coordinate ascent towards a local minimum. First, using the method of Lagrange multipliers, Eq(5) can be rewritten as the following:

$$\min_{\mathbf{a},\mathbf{w},b,\epsilon} Q = \min_{\mathbf{a},\mathbf{w},b,\epsilon}\max_{\alpha,\mu}\left\{\text{tr}(\mathbf{a}^T XLX^T\mathbf{a}) + \gamma[\frac{1}{2}||\mathbf{w}||_2^2 - F^T(X_L^T\mathbf{a}\mathbf{w} - B) + M]\right\}, s.t. \mathbf{a}^T XDX^T\mathbf{a} = I.$$

where we have defined quantities: $F = [\alpha_1 y_1, \ldots, \alpha_{n_L} y_{n_L}]^T$, $B = [b, \ldots, b]^T$, $M = C\sum_{i=1}^{n_L}\epsilon_i + \sum_{i=1}^{n_L}\alpha_i(1-\epsilon_i) - \sum_{i=1}^{n_L}\mu_i\epsilon_i$, and non-negative Lagrange multipliers $\alpha = \alpha_1, \ldots, \alpha_{n_L}$, $\mu = \mu_i, \ldots, \mu_{n_L}$. By differentiating $Q$ with respect to $\mathbf{w}$, $b$, $\epsilon_i$ we get:

$$\frac{\partial Q}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum\nolimits_{i=1}^{n_L}\alpha_i y_i \mathbf{a}^T\mathbf{x}_i = \mathbf{a}^T X_L F\ . \quad (6)$$

$$\frac{\partial Q}{\partial b} = 0 \Rightarrow \sum\nolimits_{i=1}^{n_L}\alpha_i y_i = 0, \quad \frac{\partial Q}{\partial \epsilon_i} = 0 \Rightarrow C - \alpha_i - \mu_i = 0\ . \quad (7)$$

Note Eq(6) and Eq(7) are the same as those seen in SVM optimization [3], with the only difference that the data points are now transformed by $\mathbf{a}$ as $\tilde{\mathbf{x}}_i = \mathbf{a}^T\mathbf{x}_i$. That is, given a known $\mathbf{a}$, the optimal $\mathbf{w}$ can be obtained through the standard SVM optimization process. Secondly, by substituting Eq(6) into Eq(5), we get:

$$\min_{\mathbf{a}} Q = \min_{\mathbf{a}} \Big\{ \mathrm{tr}(\mathbf{a}^T X L X^T \mathbf{a}) + \frac{\gamma}{2} F^T X_L^T \mathbf{a}\mathbf{a}^T X_L F \Big\}, \quad s.t.\ \mathbf{a}^T X D X^T \mathbf{a} = I \ . \tag{8}$$

$$\frac{\partial Q}{\partial \mathbf{a}} = 0 \quad \Rightarrow \ (XLX^T + \frac{\gamma}{2} X_L F F^T X_L^T)\mathbf{a} = \lambda X D X^T \mathbf{a} \ . \tag{9}$$

It is easy to see that $XLX^T + \frac{\gamma}{2} X_L F F^T X_L^T$ is positive semi-definite and we can update $\mathbf{a}$ by solving the generalized eigenvalue problem described in Eq(9).

Combining the above two components, we have a two-step interative process to optimize the combined cost function:

**Step-1.** With the current projection matrix $\mathbf{a}_t$ at the $t$-th iteration, train an SVM classifier to get $\mathbf{w}_t$ and $\alpha_{1,t}, \ldots, \alpha_{n_L,t}$.

**Step-2.** With the current $\mathbf{w}_t$ and $\alpha_{1,t}, \ldots, \alpha_{n_L,t}$, update the projection matrix $\mathbf{a}_{t+1}$ by solving the generalized eigenvalue problem in Eq(9).

### 3.2 Kernel LPSSVM

In this section, we show that the LPSSVM method proposed above can be extended to a nonlinear kernel version. Assume that $\phi(\mathbf{x}_i)$ is the projection function which maps the original data point $\mathbf{x}_i$ into a high-dimension feature space. Similar to the approach used in Kernel PCA [14] or Kernel LPP [9], we pursue the projection matrix $\mathbf{a}$ in the span of existing data points, i.e.,

$$\mathbf{a} = \sum\nolimits_{i=1}^{n} \phi(\mathbf{x}_i) v_i = \phi(X)\mathbf{v} \ . \tag{10}$$

where $\mathbf{v} = [v_1, \ldots, v_n]^T$. Let $K$ denote the kernel matrix over the entire data set $X = [X_L, X_U]$, where $K_{ij} = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$. $K$ can be written as: $K = \begin{bmatrix} K_L & K_{LU} \\ K_{UL} & K_U \end{bmatrix}$, where $K_L$ and $K_U$ are the kernel matrices over the labeled subset $X_L$ and the unlabeled subset $X_U$ respectively; $K_{LU}$ is the kernel matrix between the labeled data set and the unlabeled data set and $K_{UL}$ is the kernel matrix between the unlabeled data and the labeled data ($K_{LU} = K_{UL}^T$).

In the kernel space, the projection updating equation (i.e., Eq(8)) turns to:

$$\min_{\mathbf{a}} Q = \min_{\mathbf{a}} \Big\{ \mathrm{tr}(\mathbf{a}^T \phi(X) L \phi^T(X) \mathbf{a}) + \frac{\gamma}{2} F^T \phi^T(X_L) \mathbf{a}\mathbf{a}^T \phi(X_L) F \Big\}, s.t. \mathbf{a}^T \phi(X) D \phi^T(X) \mathbf{a} = I \ .$$

By differentiating $Q$ with respect to $\mathbf{a}$, we can get:

$$\phi(X) L \phi^T(X)\mathbf{a} + \frac{\gamma}{2} \phi(X_L) F F^T \phi^T(X_L)\mathbf{a} = \lambda \phi(X) D \phi^T(X) \mathbf{a}$$
$$\Rightarrow \ \Big( KLK + \frac{\gamma}{2} K^{LU|L} F F^T (K^{LU|L})^T \Big) \mathbf{v} = \lambda K D K \mathbf{v} \ . \tag{11}$$

where $K^{LU|L}=[K_L^T, K_{UL}^T]^T$. Eq(11) plays a role similar to Eq(9) that it can be used to update the projection matrix.

Likewise, similar to Eq(6) and Eq(7) for the linear case, we can find the maximum margin solution in the kernel space by solving the dual problem:

$$\begin{aligned}\tilde{Q}_{svm}^{dual} &= \sum\nolimits_{i=1}^{n_L} \alpha_i - \frac{1}{2}\sum\nolimits_{i=1}^{n_L}\sum\nolimits_{j=1}^{n_L} \alpha_i\alpha_j y_i y_j \phi^T(\mathbf{x}_i)\mathbf{a}\mathbf{a}^T\phi(\mathbf{x}_j) \\ &= \sum\nolimits_{i=1}^{n_L}\alpha_i - \frac{1}{2}\sum\nolimits_{i=1}^{n_L}\sum\nolimits_{j=1}^{n_L}\alpha_i\alpha_j y_i y_j \left[\sum\nolimits_{g=1}^{n} K_{ig}^{L|LU} v_g\right]\left[\sum\nolimits_{g=1}^{n} K_{gj}^{LU|L} v_g\right] \; .\end{aligned}$$

where $K^{L|LU}=[K_L, K_{LU}]$. This is the same with the original SVM dual problem [3], except that the kernel matrix is changed from original $K$ to:

$$\hat{K} = \left[K^{L|LU}\mathbf{v}\right]\left[\mathbf{v}^T K^{LU|L}\right] \; . \tag{12}$$

Combining the above two components, we can obtain the kernel-based two-step optimization process as follows:

**Step-1:** With the current projection matrix $\mathbf{v}_t$ at iteration $t$, train an SVM to get $\mathbf{w}_t$ and $\alpha_{1,t}, \ldots, \alpha_{n_L,t}$ with the new kernel described in Eq(12).

**Step-2:** With the current $\mathbf{w}_t$, $\alpha_{1,t}, \ldots, \alpha_{n_L,t}$, update $\mathbf{v}_{t+1}$ by solving Eq(11).

In the testing stage, given a test example $\mathbf{x}_j$ ($\mathbf{x}_j$ can be an unlabeled training sample, i.e., $\mathbf{x}_j \in X_U$ or $\mathbf{x}_j$ can be an unseen test sample), the SVM classifier gives classification prediction based on the discriminative function:

$$f(\mathbf{x}_j) = \mathbf{w}^T\mathbf{a}^T\phi(\mathbf{x}_j) = \sum_{i=1}^{n_L}\alpha_i y_i \phi(\mathbf{x}_i)\mathbf{a}\mathbf{a}^T\phi(\mathbf{x}_j) = \sum_{i=1}^{n_L}\alpha_i y_i\left[\sum_{g=1}^{n} K_{ig}^{L|LU} v_g\right]\left[\sum_{g=1}^{n} K(\mathbf{x}_g, \mathbf{x}_j) v_g\right]^T .$$

Thus the SVM classification process is also similar to that of standard SVM [3], with the difference that the kernel function between labeled training data and test data is changed from $K^{L|test}$ to: $\hat{K}^{L|test} = \left[K^{L|LU}\mathbf{v}\right]\left[\mathbf{v}^T K^{LU|test}\right]$. $\mathbf{v}$ plays the role of modeling kernel-based projection $\mathbf{a}$ before computing SVM.

### 3.3 The Algorithm

The LPSSVM algorithm is summarized in Fig. 1. Experiments show usually LPSSVM converges within 2 or 3 iterations. Thus in practice we may set $T=3$. $\gamma$ controls the importance of SVM discriminative cost function in feature subspace learning. If $\gamma=0$, Eq(11) becomes traditional LPP. In experiments we set $\gamma=1$ to balance two cost components. The dimensionality of the learned feature subspace is determined by controlling the energy ratio of eigenvalues kept in solving the eigenvalue problem of Eq(11). Note that in LPSSVM, the same Gram matrix is used for both graph construction and SVM classification, and later (Sec.4) we will see without extensive tuning of parameters LPSSVM can get good performance. For example, the default parameter setting in LibSVM [15] may be used. This is very important in real applications, especially for large-scale image/video sets. Repeating experiments to tune parameters can be time and resource consuming.

```
Input: $n_L$ labeled data $X_L$, and $n_U$ unlabeled data $X_U$.
1 Choose a kernel function $K(x,y)$, and compute Gram matrix $K_{ij}=K(\mathbf{x}_i,\mathbf{x}_j)$, e.g. RBF kernel $K(\mathbf{x}_i,\mathbf{x}_j)=\exp\{-\theta||\mathbf{x}_i-\mathbf{x}_j||_2^2\}$ or Spatial Pyramid Match Kernel [16].
2 Construct data adjacency graph over entire $X_L \cup X_U$ using $kn$ nearest neighbors. Set edge weights $W_{ij}$ based on the kernel matrix described in step 1.
3 Compute graph Laplacian matrix: $L=D-W$ where $D$ is diagonal, $D_{ii}=\sum_j W_{ij}$.
4 Initialization: train SVM over Gram matrix of labeled $X_L$, get $\mathbf{w}_0$ and $\alpha_{1,0},\ldots,\alpha_{n_L,0}$.
5 Iteration: for $t=1,\ldots,T$
    – Update $\mathbf{v}_t$ by solving problem in Eq(11) with $\mathbf{w}_{t-1}$ and $\alpha_{1,t-1},\ldots,\alpha_{n_L,t-1}$.
    – Calculate new kernel by Eq(12) using $\mathbf{v}_t$. Train SVM to get $\mathbf{w}_t$, $\alpha_{1,t},\ldots,\alpha_{n_L,t}$.
    – Stop iteration if $\sum_{i=1}^{n_L}(\alpha_{i,t-1}-\alpha_{i,t})^2<\tau$.
```

**Fig. 1.** The LPSSVM algorithm

In terms of speed, LPSSVM is very fast in the testing stage, with complexity similar to that of standard SVM classification. In training stage, both steps of LPSSVM are fast. The generalized eigenvalue problem in Eq(11) has a time complexity of O($n^3$) ($n=n_L+n_U$). It can be further reduced by exploiting the sparse implementation of [17]. For step 1, the standard quadratic programming optimization for SVM is O($n_L^3$), which can be further reduced to linear complexity (about O($n_L$)) by using efficient solvers like [18].

## 4 Experiments

We conduct experiments over 4 data sets: a toy set, two UCI sets [11], Caltech 101 for image classification [12], and Kodak's consumer video set for concept detection [13]. We compare with some state-of-the-arts, including supervised SVM [3], semi-supervised LapSVM and LapRLS [10]. We also compare with a naive LPP+SVM: first apply kernel-based LPP to get projection and then learn SVM in projected space. For fair comparison, all SVMs in different algorithms use RBF kernels for classifying UCI data, Kodak's consumer videos, and toy data, and use the Spatial Pyramid Match (SPM) kernel [16] for classifying Caltech 101 (see Sec.4.3 for details). This is motivated by the promising performance in classifying Caltech 101 in [16] by using SPM kernels. In LPSSVM, $\gamma=1$ in Eq(5) to balance the consideration on discrimination and smoothness, and $\theta=1/d$ in RBF kernel where $d$ is feature dimension. This follows the suggestion of the popular toolkit LibSVM [15]. For all algorithms, the error control parameter $C=1$ for SVM. This parameter setting is found robust for many real applications [15]. Other parameters: $\gamma_A, \gamma_I$ in LapSVM, LapRLS [10] and $kn$ for graph construction, are determined through cross validation. LibSVM [15] is used for SVM, and source codes from [17] is used for LPP.

### 4.1 Performance over Toy Data

We construct a "three suns" toy problem in Fig. 2. The data points with each same color (red, blue or cyan) come from one category, and we want to separate

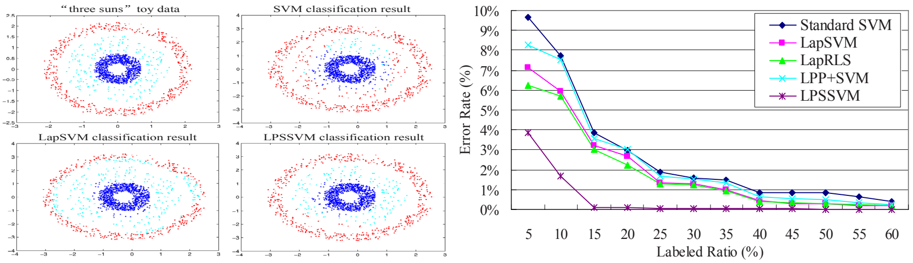


**Fig. 2.** Performance over toy data. Compared with others, LPSSVM effectively discriminates 3 categories. Above results are generated by using the SVM Gram matrix directly for constructing Laplacian graph. With more deliberate tuning of the Laplacian graph, LapSVM, LapRLS, and LPSSVM can give better results. Note that the ability of LPSSVM to maintain good performance without graph tuning is important.

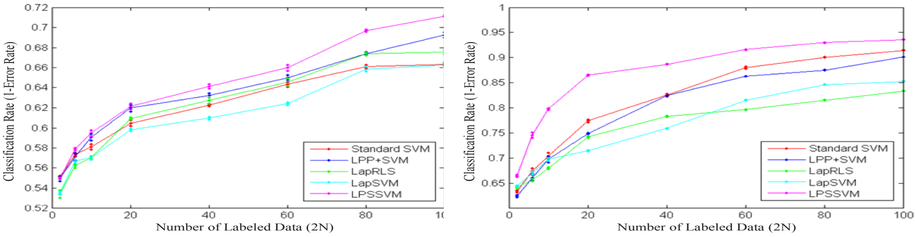
the three categories. This data set is hard since data points around the class boundaries from different categories (red and cyan, and blue and cyan) are close to each other. This adds great difficulty to manifold learning. The one-vs.-all classifier is used to classify each category from others, and each test data is assigned the label of the classifier with the highest classification score. Fig. 2 gives an example of the classification results using different methods with 10% samples from each category as labeled data (17 labeled samples in total). The averaged classification error rates (over 20 randomization runs) when varying the number of labeled data are also shown. The results clearly show the advantage of our LPSSVM in discriminative manifold learning and classifier learning.

### 4.2 Performance over UCI Data

This experiment is performed on two UCI data sets [11]: Johns Hopkins Ionosphere (351 samples with 34-dimension features), and Sonar (208 samples with 60-dimension features). Both data sets are binary classification problems. In Fig. 3 we randomly sample $N$ points from each category ($2N$ points in total) as labeled data and treat the rest data as unlabeled data as well as test data for evaluation. The experiments are repeated for 20 randomization runs, and the averaged classification rates (1 - error rates) are reported. From the result, our LPSSVM consistently outperforms all other competing methods over different numbers of labeled data in both data sets.

### 4.3 Performance over Caltech 101

The Caltech 101 set [12] consists of images from 101 object categories and an additional background class. This set contains some variations in color, pose and lighting. The bag-of-features representation [19] with local SIFT descriptors [20] has been proven effective for classifying this data set by previous works [16]. In this paper we adopt the SPM approach proposed in [16] to measure the image similarity and compute the kernel matrix. In a straightforward implementation



(a) Sonar

(b) Johns Hopkins Ionosphere

**Fig. 3.** Classification rates over UCI data sets. The *vertical dotted line over each point* shows the standard deviation over 20 randomization runs.

of SPM, only the labeled data is fed to the kernel matrix for standard SVM. For other methods, the SPM-based measure is used to construct kernel matrices for both labeled and unlabeled data (i.e., $K_L, K_U, K_{LU}$) before various semi-supervised learning methods are applied. Specifically, for each image category, 5 images are randomly sampled as labeled data and 25 images are randomly sampled as unlabeled data for training. The remaining images are used as novel test data for evaluation (we limit the maximum number of novel test images in each category to be 30). Following the procedure of [16], a set of local SIFT features of 16 × 16 pixel patches are uniformly sampled from these images over a grid with spacing of 8 pixels. Then for each image category, a visual codebook is constructed by clustering all SIFT features from 5 labeled training images into 50 clusters (codewords). Local features in each image block are mapped to the codewords to compute codeword histograms. Histogram intersections are calculated at various locations and resolutions (2 levels), and are combined to estimate similarity between image pairs. One-vs.-all classifiers are built for classifying each image category from the other categories, and a test image is assigned the label of the classifier with the highest classification score.

Table 1 (a) and (b) give the average recognition rates of different algorithms over 101 image categories for the unlabeled data and the novel test data, respectively. From the table, over the unlabeled training data LPSSVM can improve baseline SVM by about 11.5% (on a relative basis). Over the novel test data, LPSSVM performs quite similarly to baseline SVM[1].

It is interesting to notice that all other competing semi-supervised methods, i.e., LapSVM, LapRLS, and naive LPP+SVM, get worse performance than LPSSVM and SVM. Please note that extensive research has been conducted for supervised classification of Caltech 101, among which SVM with SPM kernels gives one of the top performances. To the best of our knowledge, there is no report showing that the previous semi-supervised approaches can compete this state-of-the-art SPM-based SVM in classifying Caltech 101. The fact that our LPSSVM can outperform this SVM, to us, is very encouraging.

[1] Note the performance of SPM-based SVM here is lower than that reported in [16]. This is due to the much smaller training set than that in [16]. We focus on scenarios of scarce training data to access the power of different semi-supervised approaches.

**Table 1.** Recognition rates for Caltech 101. All methods use SPM to compute image similarity and kernel matrices. Numbers shown in parentheses are standard deviations.

(a) Recognition rates (%) over unlabeled data

| SVM | LapSVM | LapRLS | LPP+SVM | LPSSVM |
|---|---|---|---|---|
| 30.2(±0.9) | 25.1(±1.1) | 28.6(±0.8) | 14.3(±4.7) | 33.7(±0.8) |

(b) Recognition rates (%) over novel test data

| SVM | LapSVM | LapRLS | LPP+SVM | LPSSVM |
|---|---|---|---|---|
| 29.8(±0.8) | 24.5(±0.9) | 26.1(±0.8) | 11.7(±3.9) | 30.1(±0.7) |

The reason other competing semi-supervised algorithms have a difficult time in classifying Caltech 101 is because of the difficulty in handling small sample size in high dimensional space. With only 5 labeled and 25 unlabeled high dimensional training data from each image category, curse of dimensionality usually hurts other semi-supervised learning methods as the sparse data manifold is difficult to learn. By simultaneously discovering lower-dimension subspace and balancing class discrimination, our LPSSVM can alleviate this small sample learning difficulty and achieve good performance for this challenging condition.

### 4.4 Performance over Consumer Videos

We also use the challenging Kodak's consumer video data set provided in [13], [21] for evaluation. Unlike the Caltech images, content in this raw video source involves more variations in imaging conditions (view, scale, lighting) and scene complexity (background and number of objects). The data set contains 1358 video clips, with lengths ranging from a few seconds to a few minutes. To avoid shot segmentation errors, keyframes are sampled from video sequences at a 10-second interval. These keyframes are manually labeled to 21 semantic concepts. Each clip may be assigned to multiple concepts; thus it represents a multi-label corpus. The concepts are selected based on actual user studies, and cover several categories like activity, occasion, scene, and object.

To explore complementary features from both audio and visual channels, we extract similar features as [21]: visual features, e.g., grid color moments, Gabor texture, edge direction histogram, from keyframes, resulting in 346-dimension visual feature vectors; Mel-Frequency Cepstral Coefficients (MFCCs) from each audio frame (10ms) and delta MFCCs from neighboring frames. Over the video interval associated with each keyframe, the mean and covariance of the audio frame features are computed to generate a 2550-dimension audio feature vector . Then the visual and audio feature vectors are concatenated to form a 2896-dimension multi-modal feature vector. 136 videos (10%) are randomly sampled as training data, and the rest are used as unlabeled data (also for evaluation). No videos are reserved as novel unseen data due to the scarcity of positive samples for some concepts. One-vs.-all classifiers are used to detect each concept, and average precision (AP) and mean of APs (MAP) are used as performance metrics, which are official metrics for video concept detection [22].

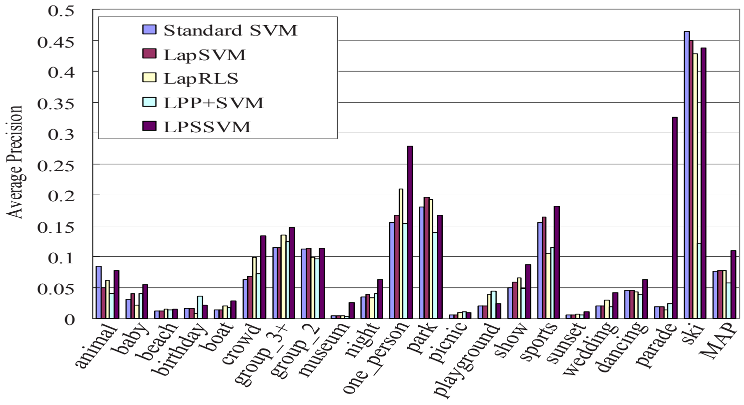
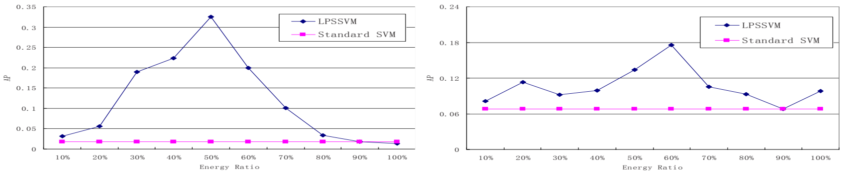


**Fig. 4.** Performance over consumer videos: per-concept AP and MAP. LPSSVM gets good performance over most concepts with strong cues from both visual and audio channels, where LPSSVM can find discriminative feature subspaces from multi-modalities.

Fig. 4 gives the per-concept AP and the overall MAP performance of different algorithms[2]. On average, the MAP of LPSSVM significantly outperforms other methods - 45% better than the standard SVM (on a relative basis), 42%, 41% and 92% better than LapSVM, LapRLS and LPP+SVM, respectively. From Fig. 4, we notice that our LPSSVM performs very well for the "parade" concept, with a 17-fold performance gain over the 2nd best result. Nonetheless, even if we exclude "parade" and calculate MAP over the other 20 concepts, our LPSSVM still does much better than standard SVM, LapSVM, LapRLS, and LPP+SVM by 22%, 15%, 18%, and 68%, respectively.

Unlike results for Caltech 101, here semi-supervised LapSVM and LapRLS also slightly outperform standard SVM. However, the naive LPP+SVM still performs poorly - confirming the importance of considering subspace learning and discriminative learning simultaneously, especially in real image/video classification. Examining individual concepts, LPSSVM achieves the best performance for a large number of concepts (14 out of 21), with a huge gain (more than 100% over the 2nd best result) for several concepts like "boat", "wedding", and "parade". All these concepts generally have strong cues from both visual and the audio channels, and in such cases LPSSVM takes good advantage of finding a discriminative feature subspace from multiple modalities, while successfully harnessing the challenge of the high dimensionality associated with the multimodal feature space. As for the remaining concepts, LPSSVM is 2nd best for 4 additional concepts. LPSSVM does not perform as well as LapSVM or LapRLS for the rest 3 concepts (i.e., "ski", "park", and "playground"), since there are no consistent audio cues associated with videos in these classes, and thus it is difficult to learn an effective feature subspace. Note although for "ski" visual

[2] Note the SVM performance reported here is lower than that in [21]. Again, this is due to the much smaller training set than that used in [21].



(a) parade (b) crowd

**Fig. 5.** Effect of varying energy ratio (subspace dimensionality) on the detection performance. There exists a reasonable range of energy ratio that LPSSVM performs well.

features have consistent patterns, the performance may be influenced more by high-dimension audio features than by visual features.

Intriguing by the large performance gain for several concepts like "parade", "crowd", and "wedding", we analyze the effect of varying dimensionality of the subspace on the final detection accuracy. The subspace dimensionality is determined by the energy ratio of eigenvalues kept in solving the generalized eigenvalue problem. As shown in Fig. 5, even if we keep only 10% energy, LPSSVM still gets good performance compared to standard SVM - 73% gain for "parade" and 20% gain for "crowd". On the other hand, when we increase the subspace dimensionality by setting a high energy ratio exceeding 0.7 or 0.8, the performances start to decrease quickly. This further indicates that there exist effective low-dimension manifolds in high-dimension multi-modal feature space, and LPSSVM is able to take advantage of such structures. In addition, there exists a reasonable range of energy ratio (subspace dimension) that LPSSVM will outperform competing methods. How to automatically determine subspace dimension is an open issue and will be our future work.

## 5 Conclusion

We propose a novel learning framework, LPSSVM, and optimization methods for tackling one of the major barriers in large-scale image/video concept classification - combination of small training size and high feature dimensionality. We develop an effective semi-supervised learning method for exploring the large amount of unlabeled data, and discovering subspace structures that are not only suitable for preserving local neighborhood smoothness, but also for discriminative classification. Our method can be readily used to evaluate unseen test data, and extended to incorporate nonlinear kernel formulation. Extensive experiments are conducted over four different types of data: a toy set, two UCI sets, the Caltech 101 set and the challenging Kodak's consumer videos. Promising results with clear performance improvements are achieved, especially under adverse conditions of very high dimensional features with very few training samples where the state-of-the-art semi-supervised methods generally tend to suffer.

Future work involves investigation of automatic determination of the optimal subspace dimensionality (as shown in Fig. 5). In addition, there is another way to optimize the proposed joint cost function in Eq(5). With relaxation $\mathbf{a}^T XDX^T \mathbf{a} - I \succeq 0$ instead of $\mathbf{a}^T XDX^T \mathbf{a} - I = 0$, the problem can be solved via SDP (Semidefinite Programming), where all parameters can be recovered without resorting to iterative processes. In such a case, we can avoid the local minima, although the solution may be different from that of the original problem.

## References

1. Joachims, T.: Transductive inference for text classification using support vector machines. In: ICML, pp. 200–209 (1999)
2. Chapelle, O., et al.: Semi-supervised learning. MIT Press, Cambridge (2006)
3. Vapnik, V.: Statistical learning theory. Wiley-Interscience, New York (1998)
4. Zhu, X.: Semi-supervised learning literature survey. Computer Sciences Technique Report 1530. University of Wisconsin-Madison (2005)
5. Bengio, Y., Delalleau, O., Roux, N.: Efficient non-parametric function induction in semi-supervised learning. Technique Report 1247, DIRO. Univ. of Montreal (2004)
6. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15, 1373–1396 (2003)
7. Cai, D., et al.: Spectral regression: a unified subspace learning framework for content-based image retrieval. ACM Multimedia (2007)
8. Duda, R.O., et al.: Pattern classification, 2nd edn. John Wiley and Sons, Chichester (2001)
9. He, X., Niyogi, P.: Locality preserving projections. Advances in NIPS (2003)
10. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of Machine Learning Research 7, 2399–2434 (2006)
11. Blake, C., Merz, C.: Uci repository of machine learning databases (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html
12. Li, F., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: CVPR Workshop on Generative-Model Based Vision (2004)
13. Loui, A., et al.: Kodak's consumer video benchmark data set: concept definition and annotation. In: ACM Int'l Workshop on Multimedia Information Retrieval (2007)
14. Schölkopf, B., Smola, A., Müller, K.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10, 1299–1319 (1998)
15. Hsu, C., Chang, C., Lin, C.: A practical guide to support vector classification, http://www.csie.ntu.edu.tw/~cjlin/libsvm/
16. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR, vol, 2, pp. 2169–2178
17. Cai, D., et al.: http://www.cs.uiuc.edu/homes/dengcai2/Data/data.html
18. Joachims, T.: Training linear svms in linear time. ACM KDD, 217–226 (2006)
19. Fergus, R., et al.: Object class recognition by unsupervised scale-invariant learning. In: CVPR, pp. 264–271 (2003)
20. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110 (2004)
21. Chang, S., et al.: Large-scale multimodal semantic concept detection for consumer video. In: ACM Int'l Workshop on Multimedia Information Retrieval (2007)
22. NIST TRECVID (2001 – 2007), http://www-nlpir.nist.gov/projects/trecvid/