SP-SVM: Large Margin Classifier for Data on Multiple Manifolds

Bin Shen[⋆], **Bao-Di Liu**[†], **Qifan Wang**[⋆], **Yi Fang**[‡], **Jan P. Allebach**[◇]
[⋆]Department of Computer Science, Purdue University, West Lafayette, IN. 47907, USA
[†]College of Information and Control Engineering, China University of Petroleum, Qingdao 266580, China
[‡]Department of Computer Engineering, Santa Clara University, Santa Clara, CA. 95053, USA
[◇]School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN. 47907, USA
bshen@purdue.edu, thu.liubaodi@gmail.com, wang868@purdue.edu, yfang@scu.edu, allebach@ecn.purdue.edu

Abstract

As one of the most important state-of-the-art classification techniques, Support Vector Machine (SVM) has been widely adopted in many real-world applications, such as object detection, face recognition, text categorization, etc., due to its competitive practical performance and elegant theoretical interpretation. However, it treats all samples independently, and ignores the fact that, in many real situations especially when data are in high dimensional space, samples typically lie on low dimensional manifolds of the feature space and thus a sample can be related to its neighbors by being represented as a linear combination of other samples on the same manifold. This linear representation, which is usually sparse, reflects the structure of underlying manifolds. It has been extensively explored in the recent literature and proven to be critical for the performance of classification. To benefit from both the underlying low dimensional manifold structure and the large margin classifier, this paper proposes a novel method called Sparsity Preserving Support Vector Machine(SP-SVM), which explicitly considers the sparse representation of samples while maximizing the margin between different classes. Consequently, SP-SVM inherits both the discriminative power of support vector machine and the merits of sparsity. A set of experiments on real-world benchmark data sets show that SP-SVM achieves significantly higher precision on recognition task than various competitive baselines including the traditional SVM, the sparse representation based method and the classical nearest neighbor classifier.

Introduction

Pattern classification is one of the most important problems in machine learning, and Support Vector Machine(SVM) (Vapnik 1963; 1998) is one of the most successful classification approaches. It has been widely used in many real-world applications, such as face recognition (Guo, Li, and Chan 2000), object detection (Felzenszwalb et al. 2010), text categorization (Joachims 1998), image classification (Liu et al. 2014; Wang, Ruan, and Si 2014; Shen, Liu, and Allebach 2014), and so on. In many of these applications, SVM is able to achieve very competitive performance compared with other approaches. Conventional SVM learns a classifier in a fully supervised manner. Given a set of labeled data samples, it aims to find the hyperplane that separates samples of different labels with the largest margin. Later, it is extended to the semi-supervised fashion (Belkin, Niyogi, and Sindhwani 2006). This semi-supervised version of SVM tries to preserve the neighborhood relation among the unlabeled samples while searching for the separating hyperplane for those labeled samples. In this manner, both labeled and unlabeled samples contribute to the final solution. However, it does not explore the neighborhood relation or local structure among labeled samples, thus ignores the underlying manifold structure of labeled samples. Furthermore, maximum margin clustering (Xu et al. 2004), which can be viewed as an unsupervised counterpart of SVM, tries to find hyperplane which separates the unlabeled samples with the largest margin. Recently, with the rapidly increasing size of data the speed of training becomes a concern for users. To speed up the training process of SVM and deal with large scale data, various optimization approaches are proposed, such as Pegasos (Shalev-Shwartz, Singer, and Srebro 2007), LIBLINEAR (Fan et al. 2008), and StreamSVM (Matsushima, Vishwanathan, and Smola 2012). With the help of these techniques, SVM is able to cope with large scale data in relatively short time.

In many applications such as face recognition, while the original data samples seem in a very high-dimensional feature space, they often actually lie on lower dimensional manifolds of the feature space (Arandjelovic et al. 2005; Shen and Si 2010; Wu, Shen, and Ling 2014; Liu et al. 2013). A reasonable assumption is that a sample can be represented as a sparse linear combination of other samples on the same manifold. For classification, the samples are drawn from different classes, which are typically on different manifolds. For example, in face recognition, the face images of the same person are on the same manifold while faces of different people lie on different manifolds. A more subtle assumption is that samples within the same class may still lie on a mixture of manifolds. For example, the face images of the same person with different facial expressions but the same lighting condition are on one manifold, while the images with the same facial expression but different lighting conditions are on another manifold. Because of the underlying low dimensional manifold structure, the samples usually

have sparse representations with respect to certain dictionary of base elements.

In recent years, sparse representation has been a surge of interest in many communities including computer vision and signal processing (Donoho 2006; Candes, Romberg, and Tao 2006; Elhamifar and Vidal 2009). Each sample can often be represented as a sparse linear combination of other samples, and this linear representation is believed to be critical for classification of high dimensional data samples (Wright et al. 2009) such as face images. Moreover, even if the linear combination is not sparse, it can also help in pattern classification, which is known as collaborative representation (Zhang, Yang, and Feng 2011; Yang et al. 2012; 2013), and our formulation is applicable to this situation as well. The goal of our work is not to justify whether sparse representation or collaborative representation is better, but to consider the underlying manifold structure of training samples by incorporating linear representation into large margin classifier. Due to the popularity of sparsity, in this paper we focus on sparse representation, although our algorithm can also be applied when the representation is dense.

This paper aims to combine the discriminative power of SVM and the merits of sparse linear representation. The proposed method explicitly exploits the local sparse structure in the training samples. While seeking the separating hyperplane with large margin, the proposed algorithm preserves the local sparse linear structure, thus called Sparsity Preserving Support Vector Machine(SP-SVM).

The algorithm is evaluated on multiple real-world testbeds. It has been shown that the proposed algorithm achieves substantially improved performance in terms of precision than the traditional SVM, which benefits from exploiting the linear representation of samples.

Review of Related Techniques

In this section, we briefly review the large margin classification algorithm SVM and the concept of sparse presentation.

Support Vector Machine

Given a set of labeled samples $\{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^m$ is the feature vector of the i_{th} sample and y_i is the corresponding label(for binary classification, $y_i \in \{-1, +1\}$), SVM learns a hyperplane that maximizes the margin between samples with different labels.

In order to learn the hyperplane $w \in \mathbb{R}^m$, SVM solves the following optimization problem:

$$\min_w C \sum_i \max(0, 1 - y_i w^T x_i) + \frac{1}{2}\|w\|_2^2 \qquad (1)$$

The first term in (1) is the hinge loss, the second term is a regularizer for w, and C is a parameter controlling the tradeoff between the hinge loss and the regularization on w. The explicit bias term is omitted here to simplify the problem while keeping it entirely general since the bias could be easily introduced as an additional coordinate in the data.

Sparse Representation

In the real-world data, an observed sample $x \in R^m$ often has a sparse representation with respect to a dictionary $D \in \mathbb{R}^{m \times p}$. This indicates $x = D\beta$, where the coefficient vector $\beta \in \mathbb{R}^p$ is sparse.

In real situations, usually the observation x is polluted by noise, and thus we have $x = D\beta + \epsilon$, where $\epsilon \in \mathbb{R}^m$ denotes the noise term.

An interesting question is how to estimate the sparse representation given the dictionary D and the sample x. Lasso (Tibshirani 1996) is a popular approach for this task since $L1$ norm penalty encourages the sparsity of the representation (Donoho 2006) and it minimize the following objective function:

$$\hat{\beta} = \arg\min_\beta \frac{1}{2}\|x - D\beta\|_2^2 + \lambda\|\beta\|_1. \qquad (2)$$

Note that this is a convex optimization problem, and there exists various efficient solvers. Besides the simplicity and efficiency, Lasso is also known to be able to select the few important base elements from the dictionary D for the given sample x.

Our Approach

Here we describe the formulation of the proposed approach, which explicitly considers the sparse representation of samples on multiple manifolds while learning a large margin classifier.

Formulation

Given a set of training samples, SVM learns a separating hyperplane by maximizing the margin between samples with different labels. Even a sample is dense in the original feature space, it may still have sparse representation with respect to some dictionary of base elements. One reasonable explanation is that the samples reside on different manifolds, and a sample has a sparse representation on its manifold (Goldberg et al. 2009), such as face images. Face images of the same subject reside on the same manifold while images of different subjects reside on different manifolds. However, SVM ignores the implicit sparse representation of the samples, and consequently ignores the structure of the underlying manifold.

Since SVM is trained on samples from different classes, it is safe to assume these training samples come from different manifolds. It is believed that the success of sparse representation in face recognition benefits from this manifold structure (Wright et al. 2009).

To encode the multiple manifold structure, the implicit sparse representation of each sample with respect to the other samples in the same class is calculated. In this way, a sparse graph connecting n nodes is created to encode the manifold structure, where n is the number of samples and each node corresponds to a sample. It is worth noting that we do not simply connect neighbors according to the Euclidean distance, because even the samples in the same class may still be from different manifolds(e.g. the face images of

a person with different facial expressions but with the same lighting condition are likely to be on one manifold, while the images with the same facial expression but with different lighting conditions are on another manifold), and samples close to each other may still be on different manifolds, especially near the intersection of manifolds. Sparse representation not only weighs more on the selected samples that are close to it, but also tends to select the samples on the same manifold.

The graph created in the way described above results from the sparse representation of each sample with respect to the other samples, and thus conveys information of the underlying manifold structure. Our proposed classification algorithm incorporates a preservation term defined based on this sparse graph into the conventional SVM objective function, and thus the proposed algorithm is able to preserve the graph structure in the output space while learning a large margin classifier.

Sparse Representation for Structure Encoding

Let x_i be an arbitrary training sample and y_i be the corresponding label. x_i can be represented as a sparse linear combination of the other training samples in the same class. Let X be the matrix composed of all the training samples, i.e. $X = [x_1, x_2, ..., x_n]$, each column of this matrix is the feature vector of a training sample.

To relate sample x_i with its neighbors, we need to estimate the linear combination coefficients, which encode the structure information around sample x_i. To achieve this goal, we solve the following optimization problem:

$$\begin{aligned} S_{.i} = &\arg\min_{\beta} \frac{1}{2}\|x_i - X\beta\|_2^2 + \lambda\|\beta\|_1 \\ &s.t.\ \ \beta_i = 0 \\ &\quad \beta_j = 0,\ if\ y_j \neq y_i, \end{aligned} \tag{3}$$

where β is vector and β_k is the coefficient for sample x_k. In many cases, we are more interested in the local sparse structure. Consequently, x_i can be represented as a sparse linear combination of its neighbors. Let $\mathcal{N}(x_i)$ denote the set of neighboring samples of x_i, either k nearest neighbors or ϵ neighbors. The related optimization problem is then modified into:

$$\begin{aligned} S_{.i} = &\arg\min_{\beta} \frac{1}{2}\|x_i - X\beta\|_2^2 + \lambda\|\beta\|_1 \\ &s.t.\ \ \beta_i = 0 \\ &\quad \beta_j = 0,\ if\ y_j \neq y_i\ or\ x_j \notin \mathcal{N}(x_i). \end{aligned} \tag{4}$$

The neighboring constraint introduces locality of the sparse representation, and it also highly reduces computational cost especially when the number of training samples is large while k is relatively small.

Let $S \in \mathbb{R}^{n\times n}$ be the matrix composed of the sparse representation of all the training samples, i.e. $S = [S_{.1}, S_{.2}, ..., S_{.n}]$. It describes the geometrical information conveyed by the training data.

Sparsity Preserving Support Vector Machine

For a sample x_i with sparse representation $S_{.i}$, its identity is closely related to the samples corresponding to nonzero coefficients of $S_{.i}$.

Since the geometrical structure information encoded by the sparse representation S is critical for classification (Wright et al. 2009), we would like to preserve this while training the discriminative large margin classifier. For convenience of presentation, we introduce two new notations. Let $Y \in \mathbb{R}^{1\times n}$ be a row vector composed of the labels of all the training samples, i.e. $Y = [y_1, y_2, ..., y_n]$. Let $f(X) \in \mathbb{R}^{1\times n}$ be the output of the classifier when applied to all the training samples, i.e. $f(X) = [f(x_1), f(x_2), ..., f(x_n)]$. Specifically, for linear SVM, $f(X) = w^T X$. Again, the bias term is omitted here for convenience without sacrificing the generality of the problem.

In the input space, we have $X \approx XS$ because of the first terms in the optimization problems (3) and (4). To preserve the structure information, it is preferable to ensure $f(X) \approx f(X)S$, which can be gained by minimizing:

$$\begin{aligned} &\sum_{i=1}^{N} \|f(x_i) - f(X)S_{.i}\|_2^2 \\ =&\|f(X)(I-S)\|_2^2 \\ =&f(X)(I-S)(I-S)^T f(X)^T \\ =&w^T XTX^T w, \end{aligned} \tag{5}$$

where $S_{.i}$ is the i_{th} column of the matrix S and $T = (I - S)(I - S)^T$.

By incorporating the term above to SVM, we get the following optimization problem:

$$\begin{aligned} \min_w C\sum_i \max(0, 1 - y_i w^T x_i) + \frac{1}{2}\|w\|_2^2 \\ + \frac{\nu}{2} w^T XTX^T w. \end{aligned} \tag{6}$$

The first two terms are inherited from the conventional SVM: the first term is the hinge loss of training samples, and the second term is standard $L2$ norm regularizer. The third term tries to preserve the sparse structure information, and the parameter ν controls how much weight we put on the structure preservation. We call this technique Sparsity Preserving Support Vector Machine(SP-SVM) since it tries to preserve the structure reflected by sparse representation while learning a large margin classifier.

When ν is equal to 0, SP-SVM reduces to the conventional SVM; when C is equal to 0, it reduces to learning a mapping direction which preserves the sparse structure rather than a classifier since it does not incur any loss if a sample is wrongly classified. More interestingly, when the parameter λ in Equation (4) is equal to 0, minimizing the objective function above turns out to be learning a large margin classifier that preserves the neighborhood information.

The objective function of SP-SVM is in the form of quadratic programming, which is the same as the conventional SVM. This can be efficiently solved by standard optimization toolboxes. Also, many of the techniques (Platt and others 1998; Shalev-Shwartz, Singer, and Srebro 2007) that help to speed up the conventional SVM could be easily adapted to SP-SVM without much modification. In our implementation, we use Quasi-Newton method to minimize the objective function by relying on the subgradient.

Overall, the training process of SP-SVM involves the computation of the sparse graph S, and a quadratic programming problem. The algorithm flow is listed in the Algorithm 1. The testing process of SP-SVM is exactly the same as the conventional SVM. The label of a testing sample x is predicted as $sign(w^\top x)$, where $sign(.)$ is the simple sign function.

Algorithm 1 SP-SVM Training

Require: Data samples $X = [x_1, x_2, ..., x_n]$, sample labels $Y = [y_1, y_2, ..., y_n]$, λ, ν and k

1: **for** $i = 1; k \leq n; i{+}{+}$ **do**
2: Compute S_i according to Equation 4
3: **end for**
4: Formulate $S = [S_1, S_2, ..., S_n]$ and compute T
5: Compute the w by solving the quadratic programming problem in Equation 6
6: **return** w

Experiments

In this section, we describe the experimental settings, and show and briefly analyze the results of our proposed algorithms, sparse representation and conventional SVM.

Experimental Settings

In the experiments, three benchmark data sets including Extended Yale B database (Georghiades, Belhumeur, and Kriegman 2001), AR database (Martinez 1998), and CMU PIE database (Sim, Baker, and Bsat 2002) are used to evaluate the performance of our proposed algorithm. The images are cropped to 32×32 and power normalized (Perronnin, Sánchez, and Mensink 2010). For each data set, the data are randomly split into training set and testing set. We randomly select images per category for training and the rest for testing. Since for the face recognition task we usually have very limited training samples for each person in real situations, our experiments use 10 training samples per category. It is shown that for the Extended Yale B database and AR database, experiments with 10 training samples per category already achieve very high precision(98.21% and 97.63%); for CMU PIE database, the precision is around 90%, which allows space to improve. Then, specifically, for CMU PIE database, we conduct more experiments with more training samples per category, and the details are reported later.

To make the results more convincing, the experimental process is repeated 10 times, and the mean and standard deviation of the classification accuracy are recorded.

Sparse representation (SR) (Wright et al. 2009) explores the sparsity of the given sample, and achieves high precision in recognition, and is considered as one of the state of art approaches for face recognition. SVM is a popular and competent technique for many classification problems including face recognition. In our experiments, the proposed SP-SVM algorithm is compared with these two algorithms, as well as the classical and popular approach: nearest neighbor classification(NN).

Given a testing sample, SR calculates its sparse linear representation with respect to all the training samples by Equation 2, where x is the testing sample and D is composed of all the training samples. Feature-sign search algorithm (Lee et al. 2007) is used for solving sparse representation. Then the identity of the testing samples is determined by the fitting error. For SVM, the implementation of LIBSVM (Chang and Lin 2011) is adopted for its robustness and popularity. For SP-SVM, S is calculated according to Equation 4, where $\mathcal{N}(x_i)$ denotes the set of k nearest neighbors of x_i. One against all multi-classification strategy is adopted by both SVM and SP-SVM.

In our experiments, there are four parameters to set: C, ν, k and λ. C is varying on the grid of $\{2^0, 2^1, \cdots, 2^5\}$ for both SVM and SP-SVM. ν is varying on the grid of $\{2^{-1}, 2^0, \cdots, 2^4\}$. λ is studied on the grid of $\{10^{-4}, 10^{-3}, \cdots, 10^1\}$ for sparse representation algorithm. For k, we consider the set $\{1, 3, 5, 7, 9\}$. For the main experiments, k is set to 5, since the performance is not sensitive to k when $k \geq 3$ as we will show below.

Extended Yale B Database

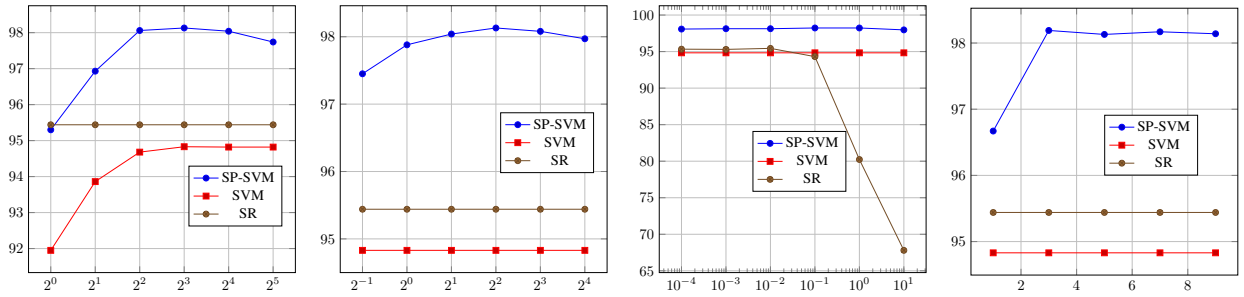
Figure 1: Examples of the Extended Yale B database

The Extended Yale B database contains 2414 frontal face images of 38 individuals, which were captured under varying illumination conditions. Figure 1 shows some sample images from this database.

Figure 2(a) shows the precision of SP-SVM, SVM, and SR with varying parameter C. As C increases, the performance of SVM and SP-SVM $\nu = 2^2$ firstly increases, till to the optimal value around $C = 2^3$, and then it decreases. From the figure, we can see that SR outperforms SVM, which does not consider the sparse representation. However, when the large margin classifier is coupled with sparsity preserving, the resulting algorithm, SP-SVM, achieves a high precision of 98.13%, which is far beyond SR(95.44%). This demonstrates the importance of the sparsity preserving. SR method considers the compact representation of the data samples and explores the underlying structure, while SVM focuses on the margin between different classes. Both

Table 1: Performance comparison on three benchmark data sets (%).

Methods	NN	SR	SVM	SP-SVM
Extended Yale B	51.45 ± 0.75	95.44 ± 0.57	94.83 ± 0.65	$\mathbf{98.13 \pm 0.46}$
CMU PIE	43.66 ± 0.88	84.55 ± 1.06	85.10 ± 0.81	$\mathbf{90.02 \pm 0.54}$
AR	49.72 ± 1.67	92.73 ± 0.33	93.67 ± 0.76	$\mathbf{97.60 \pm 0.36}$



(a) Parameter C varies ($\lambda = 0.01$, $\nu = 2^2$, $k = 5$ for SP-SVM, $\lambda = 0.01$ for SR)

(b) Parameter ν varies ($\lambda = 0.01$, $C = 2^3$, and $k = 5$ for SP-SVM, $C = 2^3$ for SVM, $\lambda = 0.01$ for SR)

(c) Parameter λ varies ($C = 2^3$, $\nu = 2^2$, and $k = 5$ for SP-SVM, $C = 2^3$ for SVM)

(d) Parameter k varies ($C = 2^3$, $\nu = 2^2$, and $\lambda = 0.01$ for SP-SVM, $C = 2^3$ for SVM, $\lambda = 0.01$ for SR)

Figure 2: Precision(%) of SP-SVM, SVM and SR with different parameter settings on Extended Yale B database

of these two facets are important for the classification, and thus when combining these two, the precision is significantly boosted.

The effect of parameter ν is also studied. We fix $C = 2^3$, and vary the value of ν. The precision(%) of SP-SVM is reported in Figure 2(b). As ν increases, the performance of SP-SVM increases, till to the optimal $\nu = 2^2$. We can also see that regardless of what value the ν is(as long as ν is in that range), the precision of SP-SVM always outperforms both SR and SVM. This further demonstrates the importance of combining of these two facets. Actually, it is not difficult to see that when $\nu = 0$, SP-SVM degenerates into the conventional SVM; when $\nu = +\infty$, the learning process of SP-SVM becomes the same as SR, though the prediction is different.

The effect of λ is shown in Figure 2(c), which shows the precision(%) of SP-SVM and SR with varying parameter λ. The precision achieved by SR algorithm is very sensitive to the value of λ, especially when $\lambda > 0.1$. When $\lambda = 10$, the precision is only 67.80%. However, the precision achieved by SP-SVM changes slowly with varying λ. Again, this insensitivity to λ should be due to the combination of the large margin and sparse representation. The effect of parameter λ for the other two data sets is nearly the same as for Extended Yale B database, i.e., the performance is not sensitive to λ, we omit these figures due to the space limit.

Figure 2(d) shows the precision(%) of SP-SVM with varying parameter k. The experiments are conducted for $k \in \{1, 3, 5, 7, 9\}$. When $k = 1$, the precision is very low. When $k \geq 3$, the precision is relatively consistent. The reason may lie in the fact that it is difficult to fit the target sample with only one neighbor. When there are more samples, sparse representation will be able to select the homogenous neighbors to fit the data point.

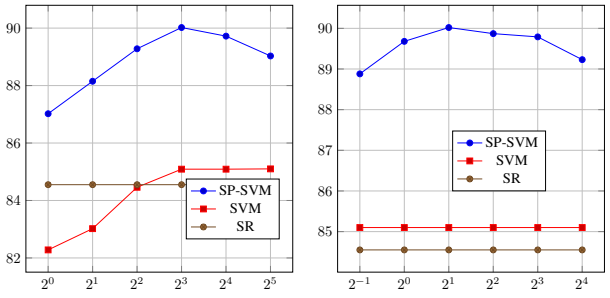
After all, the experimental results (when $\lambda = 0.01$, $C = 2^3$, $\nu = 2^2$ and $k = 5$) are listed in Table 1, which shows the precision of SP-SVM is 3.30% higher than SVM and 2.69% higher than SR, while the standard deviation of all the three algorithms is smaller than 0.7%.

CMU PIE Database

Figure 3: Examples of CMU PIE database

In the CMU PIE database, there are totally $41,368$ images of 68 people, each person under 13 different poses, 43 different illumination conditions, and with 4 different expressions. Thus, the images in the same category may still lie on multiple manifolds. We select the images with five near frontal poses (C05, C07, C09, C27, C29) and all different illuminations and expressions. There are about 170 images for each individual and totally $11,554$ images. Figure 3 shows some sample images from this database.

The results of SR, SVM, NN, and SP-SVM on PIE database are listed in Table 1 when $\lambda = 0.01$, $C = 2^3$, $\nu = 2^1$ and $k = 5$. The precision of SP-SVM is 4.92% higher than SVM and 5.47% higher than SR. Both of these gains are significant since the standard deviation is quite



(a) Parameter C varies ($\lambda = 0.01$, $\nu = 2^1$, $k = 5$ for SP-SVM, $\lambda = 0.01$ for SR)

(b) Parameter ν varies ($\lambda = 0.01$, $C = 2^3$, and $k = 5$ for SP-SVM, $C = 2^5$ for SVM, $\lambda = 0.01$ for SR)

Figure 4: Precision(%) of SP-SVM, SVM and SR with different parameter settings on CMU PIE database

small. Moreover, Figure 4(a) shows the precision of SP-SVM, SVM and SR with varying parameter C, and Figure 4(b) studies the effect of varying ν.

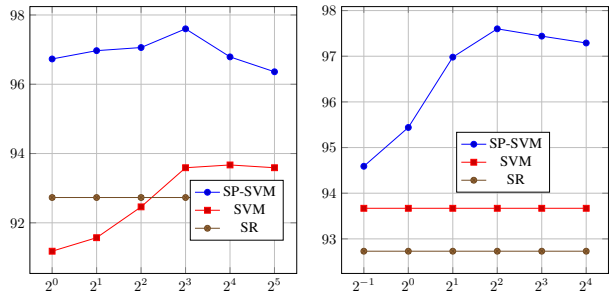
We find that for this dataset, the precision resulted from 10 training images per category is only about 90%. Then, a further experiment is conducted with 20 training images per category. With this new setting, SR achieves an average precision of 92.65% with a standard deviation of 0.53%, conventional SVM achieves 93.10% with a standard deviation of 0.42%, and the proposed SP-SVM still gets the best performance, which is 94.88% with a standard deviation of 0.22%. Although the precisions of all algorithms increase as the number of training samples increases, the proposed SP-SVM still outperforms the other two baseline algorithms substantially.

AR Database

Figure 5: Examples of AR database

There are over $4,000$ frontal images for 126 individuals in AR database. We choose a subset consisting of 50 male and 50 female categories. There are 26 images for each category. Compared with two other data sets, the AR database includes more facial variations, such as illumination change, various expressions, and facial disguises. Please refer to Figure 5 for examples.

Again, the face recognition experiments are conducted with the standard setting. The results with $\lambda = 0.01$, $C = 2^3$, $\nu = 2^2$ and $k = 5$ are reported in Table 1. The SP-SVM algorithm has much higher precision than SVM(by 3.93%) and SR (by 4.87%). Also, the standard deviation is quite small so that we can conclude that SP-SVM outperforms the others significantly. Figure 6(a) shows the precision of SP-SVM, SVM and SR with varying parameter C, and Figure 6(b) shows the precision(%) of SP-SVM with varying parameter ν with $C = 2^3$. In this wide range of C and ν, SP-SVM consistently outperforms SVM and SR.



(a) Parameter C varies ($\lambda = 0.01$, $\nu = 2^2$, $k = 5$ for SP-SVM, $\lambda = 0.01$ for SR)

(b) Parameter ν varies ($\lambda = 0.01$, $C = 2^3$, and $k = 5$ for SP-SVM, $C = 2^4$ for SVM, $\lambda = 0.01$ for SR)

Figure 6: Precision(%) of SP-SVM, SVM and SR with different parameter settings on AR database

Conclusion and Future Work

Support vector machine is one of the most popular state-of-art classification techniques. It not only has elegant theoretical interpretation, but also achieves competitive performance on many real-world tasks. Sparse representation explores the underlying manifold structure of data samples, and is proven to be critical for classification problems, especially face recognition. This paper proposes a new classification technique called SP-SVM, which combines the large margin of SVM and the power of sparse representation. The new classifier not only explores the superability of samples with different labels, but also considers the sparse relation among them, which encodes the underlying manifold structure information. Experimental results show that SP-SVM gains significantly better performance on face recognition in terms of precision than both SVM and sparse representation.

For future work, it is worth exploring to extend this sparse graph S based regularizer to semi-supervised fashion of SVM to handle a mixture of labeled and unlabeled data samples, or even the fully unsupervised version of SVM, which is maximum margin clustering. Also, another interesting direction would be to study the nonlinear version of SP-SVM, and to adopt the kernel trick into this framework.

Acknowledgements Bao-Di Liu is supported by the National Natural Science Foundation of P.R. China (No. 61402535), Qingdao Science and Technology Project (No. 14-2-4-111-jch), and the Fundamental Research Funds for the Central Universities (No. R1405012A). Yi Fang is partially supported by the TCL Research America and by the National Natural Science Foundation of China (No. 41371370).

References

Arandjelovic, O.; Shakhnarovich, G.; Fisher, J.; Cipolla, R.; and Darrell, T. 2005. Face recognition with image sets using manifold density divergence. In *IEEE Conference on Computer Vision and Pattern Recognition*.

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

Candes, E.; Romberg, J.; and Tao, T. 2006. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics* 59(8):1207–1223.

Chang, C., and Lin, C. 2011. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2(3):27.

Donoho, D. 2006. For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics* 59(6):797–829.

Elhamifar, E., and Vidal, R. 2009. Sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Fan, R.; Chang, K.; Hsieh, C.; Wang, X.; and Lin, C. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research* 9:1871–1874.

Felzenszwalb, P.; Girshick, R.; McAllester, D.; and Ramanan, D. 2010. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(9):1627 –1645.

Georghiades, A.; Belhumeur, P.; and Kriegman, D. 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(6):643–660.

Goldberg, A.; Zhu, X.; Singh, A.; Xu, Z.; and Nowak, R. 2009. Multi-manifold semi-supervised learning. In *Twelfth International Conference on Artificial Intelligence and Statistics*.

Guo, G.; Li, S.; and Chan, K. 2000. Face recognition by support vector machines. In *IEEE International Conference on Automatic Face and Gesture Recognition*.

Joachims, T. 1998. Text categorization with support vector machines: learning with many relevant features. *European Conference on Machine Learning*.

Lee, H.; Battle, A.; Raina, R.; and Ng, A. 2007. Efficient sparse coding algorithms. *Advances in Neural Information Processing Systems*.

Liu, B.-D.; Wang, Y.-X.; Zhang, Y.-J.; and Shen, B. 2013. Learning dictionary on manifolds for image classification. *Pattern Recognition* 46(7):1879–1890.

Liu, B.-D.; Wang, Y.-X.; Shen, B.; Zhang, Y.-J.; and Hebert, M. 2014. Self-explanatory sparse representation for image classification. In *European Conference on Computer Vision*. Springer. 600–616.

Martinez, A. 1998. The ar face database. *CVC Technical Report* 24.

Matsushima, S.; Vishwanathan, S.; and Smola, A. 2012. Linear support vector machines via dual cached loops. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Perronnin, F.; Sánchez, J.; and Mensink, T. 2010. Improving the fisher kernel for large-scale image classification. *European Conference on Computer Vision* 143–156.

Platt, J., et al. 1998. Sequential minimal optimization: A fast algorithm for training support vector machines. *Technical Report MSR-TR-98-14, Microsoft Research*.

Shalev-Shwartz, S.; Singer, Y.; and Srebro, N. 2007. Pegasos: Primal estimated sub-gradient solver for svm. In *International Conference on Machine Learning*, 807–814.

Shen, B., and Si, L. 2010. Non-negative matrix factorization clustering on multiple manifolds. *AAAI Conference on Artificial Intelligence*.

Shen, B.; Liu, B.-D.; and Allebach, J. P. 2014. Tisvm: Large margin classifier for misaligned image classification. *IEEE International Conference on Image Processing*.

Sim, T.; Baker, S.; and Bsat, M. 2002. The cmu pose, illumination, and expression (pie) database. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 46–51.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 267–288.

Vapnik, V. 1963. Pattern recognition using generalized portrait method. *Automation and Remote Control* 24:774–780.

Vapnik, V. 1998. Statistical learning theory.

Wang, Q.; Ruan, L.; and Si, L. 2014. Adaptive knowledge transfer for multiple instance learning in image classification. *AAAI Conference on Artificial Intelligence*.

Wright, J.; Yang, A.; Ganesh, A.; Sastry, S.; and Ma, Y. 2009. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31(2):210 –227.

Wu, Y.; Shen, B.; and Ling, H. 2014. Visual tracking via online nonnegative matrix factorization. *IEEE Transactions on Circuits and Systems for Video Technology* 24(3):374–383.

Xu, L.; Neufeld, J.; Larson, B.; and Schuurmans, D. 2004. Maximum margin clustering. *Advances in Neural Information Processing Systems* 17:1537–1544.

Yang, M.; Zhang, L.; Zhang, D.; and Wang, S. 2012. Relaxed collaborative representation for pattern classification. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Yang, W.; Wang, Z.; Yin, J.; Sun, C.; and Ricanek, K. 2013. Image classification using kernel collaborative representation with regularized least square. *Applied Mathematics and Computation* 222(0):13 – 28.

Zhang, L.; Yang, M.; and Feng, X. 2011. Sparse representation or collaborative representation: Which helps face recognition? In *IEEE International Conference on Computer Vision*.