21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

Exploiting Subclass Information in Support Vector Machines

Georgios Orfanidis, Anastasios Tefas
Dept. of Informatics, Aristotle University of Thessaloniki, 54124, Greece
tefas@aiia.csd.auth.gr

Abstract

In this paper a new variation of Support Vector Machines (SVM) is introduced. The proposed method is called Subclass Support Vector Machine (SSVM) and makes use of principles from Discriminant Analysis field using subclasses. The major difference over SVM is that it takes into account the existence of subclasses in the classes and tries to minimize the distribution of the samples within each subclass. Experiments over various databases are conducted and the results are compared against other classifiers.

1. Introduction

Support Vector Machines (SVM), [11], have been used successfully in a variety of occasions as classification tool and have been appreciated for their stability and solid theoretical foundation.

In their standard form SVM try to find a separating decision hyperplane with maximum margin between the classes [11]. One of their major advantages is their property of using a parametric technique based on structural risk analysis as opposed to nonparametric techniques. The popularity of SVM is consequence of their ability to represent each classification problem as a quadratic optimization problem. SVM are also quite popular for their ability to construct nonlinear decision surfaces. Using the Kernel trick samples are projected to a new high dimensional space where the problem can be solved.

Despite SVM strength and advantages, variations that improve some of their characteristics have been introduced [9], [12], [7]. One such variation is proposed in this paper which uses information normally neglected by SVM like division of the samples into subclasses. Both linear and non linear cases are examined and experiments are conducted.

2 Subclass Support Vector Machines

The notion of subclasses has been successfully used in the case of dimensionality reduction where the Clustering based Discriminant Analysis and the Subclass Discriminant Analysis have been proposed in [13] and [2] respectively. Moreover, in [3] subclasses have been combined with error correcting output codes in the Subclass ECOC classification framework. In all the above cases the first step is to estimate the subclasses into each class. This is achieved using a clustering algorithm like k-means, spectral clustering, Nearest Neighbor clustering or other clustering methods [10].

The clustering method used has minor impact in the classification performance as shown in [13] and thus, any clustering algorithm can work in these approaches. The optimal number of subclasses is usually chosen by a 10 fold cross validation. Similarly, the first step in the proposed SSVM is to estimate the subclasses within each class.

2.1 Linear Subclass Support Vector Machines

The subclasses found by the clustering method are being used for the calculation of the within-subclasses scatter matrix $\boldsymbol{S}_s$ proposed in [2] and defined as:

$$\boldsymbol{S}_s = \sum_{i=1}^{N_C} \sum_{j=1}^{K_i} \sum_{k=1}^{N_{ij}} p_{ij} (\boldsymbol{x}_{ijk} - \boldsymbol{\mu}_{ij})(\boldsymbol{x}_{ijk} - \boldsymbol{\mu}_{ij})^T \quad (1)$$

where N_C is the number of classes, K_i is the number of subclasses for each class i, N_{ij} is the number of samples belonging to subclass j of class i, $\boldsymbol{\mu}_{ij}$ is the mean vector of each subclass and $p_{ij} = \frac{N_{ij}}{N}$ is the prior probability of jth subclass of class i. The matrix $\boldsymbol{S}_s$ represents the dispersion within each subclass. Thus, we would like to minimize this dispersion after the projection of the initial samples to a reduced dimensionality subspace.

The proposed approach is called Subclass Support Vector Machines (SSVM) and takes into account the subclass distribution in order produce more efficient and

robust solutions than standard SVM. It is straightforward to show that the distribution of the samples in each subclass after the projection to a vector $\boldsymbol{w}$ is given by $\boldsymbol{w}^T\boldsymbol{S}_s\boldsymbol{w}$. Thus, the proposed objective criterion to be minimized is $\boldsymbol{w}^T\boldsymbol{S}_s\boldsymbol{w}$ subject to separability constraints as given in the SVM. If the samples are not linearly separable the optimization problem to be solved is (taking into account the cost of error classifications):

$$\min_{\boldsymbol{w},b} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{S}_s\boldsymbol{w} + C\sum_{i=1}^{N}\xi_i, \qquad \boldsymbol{w}^T\boldsymbol{S}_s\boldsymbol{w} > 0 \quad (2)$$

subject to the separability constraints:

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0, \quad i = 1, ..., N \quad (3)$$

where ξ_i are again non-negative slack variables used for the misclassified samples and C is a constant defining the cost of these misclassifications. By assigning values to the constant C one can reproduce the two extreme cases, one case with high punishment of errors and the second case where errors are virtually overlooked.

The proposed criterion (2) requires minimization of the samples scatter in each subclass with the constraints of SVM separability in (3). That is the solution provides well separated classes with compact subclasses.

The solution of (2) is given by the saddle point of the Lagrangian:

$$\begin{aligned} L(\boldsymbol{w}, b, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\xi}) &= \boldsymbol{w}^T\boldsymbol{S}_s\boldsymbol{w} + C\sum_{i=1}^{N}\xi_i \\ &- \sum_{i=1}^{N} a_i[y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{N}\beta_i\xi_i \end{aligned} \quad (4)$$

with $\boldsymbol{a} = [a_1, .., a_N]$ and $\boldsymbol{\beta} = [\beta_1, .., \beta_N]$ being the Lagrange multiplier vectors for the constraints imposed in (3). Provided that $\boldsymbol{S}_s$ is non-singular the optimal vector $\boldsymbol{w}$ can be found by the Karush–Kuhn–Tucker (KKT) conditions and is given by:

$$\boldsymbol{S}_s\boldsymbol{w}_o = \frac{1}{2}\sum_{i=1}^{N} a_{i,o}y_i\boldsymbol{x}_i \Leftrightarrow \boldsymbol{w}_o = \frac{1}{2}\boldsymbol{S}_s^{-1}\sum_{i=1}^{N} a_{i,o}y_i\boldsymbol{x}_i \quad (5)$$

By replacing (5) into (4) and using the KKT conditions, the optimization problem (2) is now reformulated to its dual form (offered for QP optimization):

$$\max_{\boldsymbol{a}} \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j y_i y_j K_{\text{SSVM}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$$\text{subject to } 0 \leq a_i \leq C,\ \forall i = 1, .., N \text{ and } \sum_{i=1}^{N} y_i a_i = 0 \quad (6)$$

where $\boldsymbol{K}_{\text{SSVM}} = \boldsymbol{U}^T\boldsymbol{S}_s^{-1}\boldsymbol{U}$ is the new kernel matrix created for SSVM algorithm and $\boldsymbol{U} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N] \in \Re^{DxN}$ is the initial training data matrix of the samples. For comparison reasons in standard linear SVM the kernel matrix is given by $\boldsymbol{K}_{\text{SVM}} = \boldsymbol{U}^T\boldsymbol{U}$.

2.2 Non-linear expansion of SSVM

Up to this point only linear separation surfaces have been calculated and used. It is obvious though that as with every SVM variation there is a straightforward extension from the linear classification to the non-linear one. The whole idea is based onto projecting the initial samples $\boldsymbol{x} \in \Re^D$ to a higher dimensional space $\Re^K$ with $K > D$ into which the classification problem is solved. To make the whole idea feasible the 'kernel trick' is used by replacing the inner product found in the SVM algorithm with a kernel function that uses a mapping, usually denoted as ϕ, that maps each sample from the initial space into a new arbitrary dimensional space. This new space is not necessary to be calculated explicitly but instead uses the mapping of the inner products to the new space. In this new space linear classification can be used.

After passing the samples from the mapping function we virtually have $\phi(\boldsymbol{x}) \in \Re^K$ instead of $\boldsymbol{x} \in \Re^D$. It can be pointed out that there is no bound of the dimension of the new space that could be infinite as in the case of RBF kernel functions. The relations dominating the new space are identical with the previous ones into the initial space. That is the within subclass scatter matrix is:

$$\boldsymbol{S}_s^{\phi} = \sum_{i=1}^{N_C}\sum_{j=1}^{K_i}\sum_{k=1}^{N_{ij}} p_{ij}(\phi(\boldsymbol{x}_{ijk}) - \boldsymbol{\mu}_{ij}^{\phi})(\phi(\boldsymbol{x}_{ijk}) - \boldsymbol{\mu}_{ij}^{\phi})^T \quad (7)$$

where N_C is the number of classes, K_i is the number of subclasses for each class i, N_{ij} is the number of samples belonging to subclass j of class i, $\boldsymbol{\mu}_{ij}^{\phi} = (\frac{1}{N_{ij}})\sum_{\boldsymbol{x}\in C_{ij}}\phi(\boldsymbol{x})$ is the mean vector of each subclass in the feature space, $p_{jk} = \frac{N_{jk}}{N}$ is the prior probability of the k-th subclass of class j. As can be seen the clustering is done in the initial space and only the final problem is solved in the feature space (the mapped one).

The problem now is stated as:

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{S}_s^{\phi}\boldsymbol{w} + C\sum_{i=1}^{N}\xi_i, \qquad \boldsymbol{w}^T\boldsymbol{S}_s^{\phi}\boldsymbol{w} > 0 \quad (8)$$

with constraints:

$$y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}) + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0, \quad i = 1, ..., N \quad (9)$$

which makes the analogy with the linear case obvious. A problem that is more common in the non-linear case though is the case where $\boldsymbol{S}_s^{\phi}$ is singular.

We can use an approach similar to the one proposed in [5]. That is, we can apply Singular Value Decomposition on $\boldsymbol{S}_s$ and decompose it to $\boldsymbol{S}_s = \boldsymbol{V}^T \boldsymbol{\Sigma} \boldsymbol{V}$, where $\boldsymbol{V}$ is an orthonormal matrix that contains the eigenvectors of $\boldsymbol{S}_s$ and $\boldsymbol{\Sigma}$ a diagonal matrix that contains the corresponding eigenvalues of $\boldsymbol{S}_s$. Then we can apply the transform:

$$\boldsymbol{Q} = \boldsymbol{\Sigma}^{-\frac{1}{2}} V \tag{10}$$

to the initial samples $\boldsymbol{x}_i$ calculating $\boldsymbol{x}_i^{'} = \boldsymbol{Q}\boldsymbol{x}_i$. It is straightforward to show that:

$$\boldsymbol{x}_i^{'T}\boldsymbol{x}_j^{'} = \boldsymbol{x}_i^T \boldsymbol{Q}^T \boldsymbol{Q} \boldsymbol{x}_j = \boldsymbol{x}_i^T \boldsymbol{S}_s^{-\frac{1}{2}} \boldsymbol{x}_j \tag{11}$$

The transform (11) applied to the whole initial data matrix outputs:

$$\boldsymbol{U}^{'} = [\boldsymbol{x}_1^{'}, ..., \boldsymbol{x}_N^{'}] = [\boldsymbol{Q}\boldsymbol{x}_1, ..., \boldsymbol{Q}\boldsymbol{x}_N] = \boldsymbol{Q}\boldsymbol{U} \in \Re^{D \mathrm{x} N} \tag{12}$$

and using (12) we calculate

$$\boldsymbol{U}^{'T}\boldsymbol{U}^{'} = \boldsymbol{U}^T \boldsymbol{Q}^T \boldsymbol{Q} \boldsymbol{U} = \boldsymbol{U}^T \boldsymbol{S}_s^{-\frac{1}{2}} \boldsymbol{U} \tag{13}$$

Thus, applying linear standard SVM to samples $\boldsymbol{x}_i^{'}$, i.e. using (13) as kernel matrix, corresponds to applying linear SSVM with optimization problem as stated in (6).

We can expand the above notion in non linear cases, i.e. we can use standard SVM with non linear kernels applied in the transformed samples $\boldsymbol{x}_i^{'}$ which results in applying non linear transforms on the inner products (11) that appear in the kernel matrix (13). That is, the RBF kernel can be defined as:

$$\begin{aligned} K_{SSVM-\mathrm{RBF}}(\boldsymbol{x}_i, \boldsymbol{x}_j) &= e^{-\frac{g(\boldsymbol{x}_i, \boldsymbol{x}_j)}{2\sigma^2}} \\ g(\boldsymbol{x}_i, \boldsymbol{x}_j) &= (\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{S}_s^{-\frac{1}{2}} (\boldsymbol{x}_i - \boldsymbol{x}_j) \end{aligned} \tag{14}$$

The above procedure has been used in all the experiments with non linear kernels.

3 Experimental Results

The proposed SSVM algorithm has been tested against SVM and MCVSVM from [12] to estimate its power using linear and non linear kernels. In all databases if there was a test set available performance was calculated on this set otherwise a 5×2 cross validation [1] was used. The k-means algorithm has been used in all the experiments for clustering in the SSVM framework. In the linear case the parameters $C = 10$ and $C = 100$ have been used and the results are summarized in Table 1.

Linear Classification performance			
Method →	SVM	MCVSVM	SSVM
Database ↓	$C = 10$		
ETH80	95.16%	94.78%	95.01%
Segment	87.79%	91.78%	93.15%
Shuttle	62.82%	87.74%	89.56%
Vehicle	80.71%	85.68%	86.09%
	$C = 100$		
ETH80	95.34%	95.33%	**95.54%**
Segment	90.98%	95.94%	**96.29%**
Shuttle	**90.99%**	88.39%	89.51%
Vehicle	80.71%	89.40%	**89.59%**

Table 1. Linear performance on databases for parameter $C = 10$ and $C = 100$

The experiments with non-linear kernels were made using Gaussian or Radial Basis Function kernels of the form $k(\boldsymbol{x}_i^{'}, \boldsymbol{x}_j^{'}) = e^{-\gamma \|\boldsymbol{x}_i^{'} - \boldsymbol{x}_j^{'}\|^2}$ with parameters being the parameter C (same with the parameter of linear SVM) and γ. A combination of these two parameters have been chosen to demonstrate performance of these three methods: values for C were 10 and 100 and for γ, 1 and 10. So there were four sets of experiments for all databases. Results on all databases in the non-linear case are shown in Table 2.

The first database used in the experiments is the ETH80 multi-view object recognition database [6] which consists of 8 object classes. The results displayed in Table 1 indicate better performance of SVM for parameter $C = 10$ and superiority of SSVM for parameter $C = 100$. The average number of subclasses for the 8 problems 1 vs All was for both values of C, $K1 = 8.75$ and $K2 = 21.12$, where $K1$, and $K2$ is the number of subclasses for class 1 and 2 respectively.

Non linear experiments on ETH80 database showed a better performance for standard SVM with parameter $\gamma = 1$ for both values of parameter C, $C = 10$ and $C = 100$ with the differences being -1.26% and -0.72% respectively, but also a superiority of SSVM for parameter $\gamma = 10$ also for both values of C with differences between SSVM and SVM being 1.44% and 1.14%. The overall best performance obtained by SSVM.

The second experiment includes databases that are part of the European StatLog project [4]. The 3 datasets used are Segment, Shuttle and Vehicle Silhouettes database. The image segmentation database consists of 2310 samples of 19 dimensions. Results in Table 1 showed a superiority of SSVM over SVM by nearly 5%. The average optimal number of subclasses now was for $C = 10$: $K1 = 2.57$ and $K2 = 2.42$ and

RBF Classification performance			
Method →	SVM	MCVSVM	SSVM
Database ↓	$\gamma = 1$ and $C = 10$		
ETH80	97.19%	95.23%	95.93%
Segment	91.36%	95.85%	97.51%
Shuttle	87.89%	89.44%	90.09%
Vehicle	76.54%	88.79%	90.30%
	$\gamma = 10$ and $C = 10$		
ETH80	96.12%	96.99%	**97.56%**
Segment	96.34%	97.13%	**98.28%**
Shuttle	89.73%	89.74%	**91.99%**
Vehicle	85.34%	90.92%	**92.93%**
	$\gamma = 1$ and $C = 100$		
ETH80	97.28%	96.08%	96.56%
Segment	94.76%	97.29%	98.07%
Shuttle	89.73%	89.80%	91.33%
Vehicle	85.14%	89.23%	91.87%
	$\gamma = 10$ and $C = 100$		
ETH80	96.36%	96.92%	**97.50%**
Segment	96.95%	98.21%	**98.39%**
Shuttle	90.03%	90.43%	**92.21%**
Vehicle	88.22%	90.29%	**91.88%**

Table 2. RBF performance on databases for parameter $\gamma = 1$, $\gamma = 10$, $C = 10$ and $C = 100$

for $C = 100$: $K1 = 2$ and $K2 = 1.57$ respectively. Non linear SSVM outperformed non linear SVM in all sets of parameters by a difference varying from 6.1% to 1.4%.

The vehicle silhouettes database comes from Turing Institute, Glasgow, Scotland [8]. In this database SSVM performed better than both SVM and MCVSVM by roughly 5.5% and 0.4% respectively for parameter $C = 10$ and also roughly 9% and 0.2% for parameter value $C = 100$. The average optimal number of subclasses was for $C = 10$: $K1 = 2.75$ and $K2 = 2.75$ and for $C = 100$: $K1 = 2.5$ and $K2 = 1.25$ respectively. Non linear SSVM outperformed non linear SVM in all cases with a superiority varying from 3.66% to 13.76%.

In the Shuttle database results were controversial in the linear case. For parameter $C = 10$ SSVM outperformed SVM by more than 25% but SVM proved better for $C = 100$ by nearly 1.5%. In this case the average optimal number of subclasses now was for $C = 10$: $K1 = 2.33$ and $K2 = 4$ and for $C = 100$: $K1 = 2$ and $K2 = 4.33$ respectively. Performance showed a superiority of non linear SSVM in all 4 cases by a difference approximately 1.6% to 2.2%.

4 Conclusions

In this paper it has been shown that SSVM is a new promising method that uses notions from classic SVM and Subclass Based Discriminant Analysis in a fusion way expanding the idea of exploiting subclasses in SVM. Its performance on real problems proved to be quite competitive with other state of the art methods. In most cases, experiments showed an improved performance over SVM most notably in the linear case. Thus, we may conclude that SSVM achieved separability improvement on the standard SVM.

References

[1] E. Alpaydin. Combined 5 x 2 cv f test for comparing supervised classification learning algorithms. *Neural Comput.*, 11:1885–1892, November 1999.

[2] X. Chen and T. Huang. Facial expression recognition: A clustering-based approach. *Pattern Recognition Letters*, 24:1295–1302, 2003.

[3] S. Escalera, D. Tax, O. Pujol, P. Radeva, and R. Duin. Subclass problem-dependent design of error-correcting output codes. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, 30:1041–1054, June 2008.

[4] C. Feng, A. Sutherland, S. King, and S. Muggleton. Comparison of machine learning classifiers to statistics and neural networks. *AI and Stats Conf.*, 1993.

[5] T. S. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 487–493, 1999.

[6] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. *International Conference on Computer Vision and Pattern Recognition CVPR '03*, June 2003.

[7] P. Shivaswamy and T. Jebara. Relative margin machines. *Neural Information Processing Systems, NIPS*, 2008.

[8] J. Siebert. Vehicle recognition using rule based methods. *Turing Institute Research Memorandum TIRM-87-018*, March 1987.

[9] A. Tefas, C. Kotropoulos, and I. Pitas. Using support vector machines to enhance the performance of elastic graph matching for frontal face authentication. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(7):735–746, July 2001.

[10] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, November 2008.

[11] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, NY, 1998.

[12] S. Zafeiriou, A. Tefas, and I. Pitas. Minimum class variance support vector machines. *IEEE Transaction on image processing*, 10:2551–2564, October 2007.

[13] M. Zhu and A. M. Martinez. Subclass discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8), August 2006.