Unsupervised and Semi-supervised Lagrangian Support Vector Machines *

Kun Zhao[1], Ying-Jie Tian[2], and Nai-Yang Deng[1,**]

[1] College of Science
China Agricultural University
piaopiao-zk@163.com,dengnaiyang@vip.163.com
[2] Chinese Academy of Sciences
Research Center on Data Technology and Knowledge Economy
tianyingjie1213@163.com

Abstract. Support Vector Machines have been a dominant learning technique for almost ten years, moreover they have been applied to supervised learning problems. Recently two-class unsupervised and semi-supervised classification problems based on Bounded C-Support Vector Machines and Bounded ν-Support Vector Machines are relaxed to semi-definite programming[4][11]. In this paper we will present another version to unsupervised and semi-supervised classification problems based on Lagrangian Support Vector Machines, which trained by convex relaxation of the training criterion: find a labelling that yield a maximum margin on the training data. But the problems have difficulty to compute, we will find their semi-definite relaxations that can approximate them well. Experimental results show that our new unsupervised and semi-supervised classification algorithms often obtain almost the same accurate results as the unsupervised and semi-supervised methods [4][11], while considerably faster than them.

Keywords: Lagrangian Support Vector Machines, Semi-definite Programming, unsupervised learning, semi-supervised learning, margin.

1 Introduction

As an important branch in unsupervised learning, clustering analysis aims at partitioning a collection of objects into groups or clusters so that members within each cluster are more closely related to one another than objects assigned to different clusters[1]. Clustering algorithms provide automated tools to help identify a structure from an unlabelled set, in a variety of areas including bio-informatics, computer vision, information retrieval and data mining. There is a rich resource of prior works on this subject. The works reviewed below are most related to ours.

* This work is supported by the National Natural Science Foundation of China (No. 10371131,10631070 and 10601064).

** The corresponding author.

Y. Shi et al. (Eds.): ICCS 2007, Part III, LNCS 4489, pp. 882–889, 2007.

Efficient convex optimization techniques have had a profound impact on the field of machine learning. Most of them have been used in applying quadratic programming techniques to Support Vector Machines (SVMs) and kernel machine training[2]. Semi-definite Programming (SDP) extends the toolbox of optimization methods used in machine learning, beyond the current unconstrained, linear and quadratic programming techniques.

Semi-definite Programming (SDP) has showed its utility in machine learning. Lanckreit *et al* show how the kernel matrix can be learned from data via semi-definite programming techniques[3]. De Bie and Cristanini develop a new method for two-class transduction problem based on semi-definite relaxation technique[5]. Xu *et al* based on[3][5] develop methods to two-class unsupervised and semi-supervised classification problems in virtue of relaxation to Semi-definite Programming[4]. Zhao *et al* present another version to unsupervised and semi-supervised classification problems based on Bounded ν-Support Vector Machines [11].

In this paper we provide a brief introduction to the application of Semi-definite Programming in machine learning[3][4][5][11] and construct other unsupervised and semi-supervised classification algorithms. They are based on Lagrangian Support Vector Machines (LSVMs), which obtain almost accurate results as other unsupervised and semi-supervised methods [4][11], while considerably faster than them.

We briefly outline the contents of the paper now. We review the Support Vector Machines and Semi-definite Programming in Section 2 . Section 3 will formulate new unsupervised and semi-supervised classification algorithms which are based on LSVMs. Experimental results will be showed in Section 4. In the last Section we will have a conclusion.

A word about our notation. All vectors will be column vectors unless transposed to a row vector by "T". The scalar (inner) product of two vectors x and y in the n-dimensional real space R^n will be denoted by $x^T y$. For an $l \times d$ matrix A, A_i will denote the ith row of A. The identity matrix in a real space of arbitrary dimension will be denoted by I, while a column vector of ones of arbitrary dimension will be denoted by e.

2 Preliminaries

Considering the supervised classification problem, we will assume the given labelled training examples $(x_1, y_1), \ldots, (x_n, y_n)$ where each example is assigned a binary $y_i \in \{-1, +1\}$. The goal of SVMs is to find the linear discriminant $f(x) = w^T \phi(x) + b$ that maximizes the minimum misclassification margin

$$\begin{aligned} \min_{w,b,\xi} \ & \frac{1}{2}(\|w\|^2 + b^2) + \frac{C}{2}\sum_{i=1}^{n} \xi_i^2 \\ \text{s.t.} \ & y_i((w \cdot \phi(x_i)) - b) + \xi_i \geq 1 \ , i = 1, 2, \ldots, n \end{aligned} \tag{1}$$

Let $\Phi = (\phi(x_1), \ldots, \phi(x_n))$, $K = \Phi^T\Phi$, then $K_{ij} = \phi(x_i)^T\phi(x_j)$, dual problem of (1) is

$$\begin{aligned} \max_{\alpha} \ & -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j K_{ij} - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j - \frac{1}{2C}\sum_{i=1}^{n}\alpha_i^2 + \sum_{i=1}^{n}\alpha_i \quad (2) \\ \text{s.t.} \ & \alpha_i \geq 0, i = 1, 2, \ldots, n \end{aligned}$$

The problem (1) and (2) are primal and dual problem of Lagrangian Support Vector Machines (LSVMs) respectively [12].

Based on Bounded C-Support Vector Machines (BC-SVMs) [8] Xu *et al* get the optimization problem [4] that can solve unsupervised classification problem. Zhao *et al* based on Bounded ν-Support Vector Machines (Bν-SVMs)[7] get the optimization problem [11] that can solve unsupervised classification problem too. In this paper Lagrangian Support Vector Machines will be used to resolve unsupervised and semi-supervised classification problems.

Given $H \in \mathcal{M}^n$, $A_i \in \mathcal{M}^n$ and $b \in \mathcal{R}^m$, where $\mathcal{M}^n$ is the set of $n \times n$ symmetric matrix. The standard Semi-definite Programming problem is to find a matrix $X \in \mathcal{M}^n$ for the optimization problem

$$\begin{aligned} & \min \ H \bullet X \\ \text{(SDP)} \qquad & \text{s.t.} \ A_i \bullet X = b_i, i = 1, 2, \ldots, m \\ & X \succeq 0 \end{aligned}$$

where the $\bullet$ operation is the matrix inner product $A \bullet B = trA^T B$, the notation $X \succeq 0$ means that X is a positive semi-definite matrix. The dual problem to (SDP) can be written as:

$$\begin{aligned} & \max \ b^T\lambda \\ \text{(SDD)} \qquad & \text{s.t.} \ H - \sum_{i=1}^{m}\lambda_i A_i \succeq 0 \end{aligned}$$

Here $\lambda \in \mathcal{R}^m$. For Semi-definite Programming, interior point method has good effect, moreover there exists several softwares such as SeDuMi[10] and SDP3.

3 Unsupervised and Semi-supervised Classification Algorithms

A recent development of convex optimization theory is Semi-definite Programming, a branch of that fields aimed at optimizing over the cone of semi-positive definite matrices. One of its main attraction is that it has been proven successful in construct tight convex relaxation of NP-hard problem. Semi-definite Programming has showed its utility in machine learning too.

Lanckreit *et al* show how the kernel matrix can be learned from data via semi-definite programming techniques[3]. They presented new methods for learning a kernel matrix from labelled data set and transductive data set. Both methods

can relax the problem to Semi-definite Programming. For a transductive setting, using the labelled data one can learn a good embedding (kernel matrix), which can then be applied to the unlabelled part of the data. De Bie and Cristanini relax two-class transduction problem to semi-definite programming based on transductive Support Vector Machines[5].

Xu *et al* develop methods to two-class unsupervised and semi-supervised classification problems based on Support Vector Machines in virtue of relaxation to Semi-definite Programming[4]in the foundation of [5][3]. Its purpose is to find a labelling which has the maximum margin not to find a large margin classifier. This leads to the method to cluster the data into two class, which subsequently run a SVM, and will obtain the maximum margin with all possible labelling. We should add constraint about class balance $-\varepsilon \leq \sum_{i=1}^{n} y_i \leq \varepsilon$, otherwise we can simply assign all the data to the same class and then get unbounded margin; moreover this can avoid noisy data's influence in some sense.

Using the method in [5][3], Xu *et al* based on BC-SVMs get the optimization problem[4]that can solve unsupervised classification problem. Analogously Zhao *et al* based on Bν-SVMs get the optimization problem[11]that can solve unsupervised classification problem too, which the parameter ν in Bν-SVMs has quantitative meaning. However, the time consumed of both methods based on BC-SVMs and Bν-SVMs is too long. So it seems necessary to find a faster method, which has almost accurate results as above at least. The reason that unsupervised classification algorithms based on BC-SVMs and Bν-SVMs run slowly is their semi-definite relaxations have so many variables, concretely n^2+2n+1 and n^2+2n+2 variables respectively. In order to fasten the speed of algorithm, it seems better to find a qualified SVM which has fewer constraints, for the number of variables in semi-definite relaxation problem equals to sum of n^2+1 and number of constraints in SVM. Primal problems of BC-SVMs and Bν-SVMs have $2n$ and $2n+1$ constraints respectively, while primal problem of Lagrangian Support Vector Machines has n constraints. Therefore it seems better to use Lagrangian Support Vector Machines to resolve unsupervised classification problem.

We use the same method in [5][3] to get the optimization problem based on LSVMs

$$\begin{aligned} \min_{y_i \in \{-1,+1\}^n} \quad & \min_{w,b,\xi} \frac{1}{2}(\|w\|^2 + b^2) + \frac{C}{2}\sum_{i=1}^{n} \xi_i^2 \\ \text{s.t.} \quad & y_i((w \cdot \phi(x_i) - b) + \xi_i \geq 1 \\ & -\varepsilon \leq \sum_{i=1}^{n} y_i \leq \varepsilon \end{aligned} \tag{3}$$

It is difficult to solve Problem (3), so we will consider to get its approximate solutions. Since Semi-definite Programming can provide effective algorithms to cope with difficult computational problems and obtain high approximate solutions, it seems better to relax problem (3) to Semi-definite Programming. Let $y = (y_1, y_2, \ldots, y_n)^T$, $M = yy^T$, $\Phi = (\phi(x_1), \ldots, \phi(x_n))$ and $K = \Phi^T\Phi$,

moreover $A \circ B$ denotes componentwise matrix multiplication. Use the same method in [3], we obtain the Unsupervised Classification Algorithm.

Algorithm 3.1 (Unsupervised Classification Algorithm)

1.Given data set $D = \{x_1, \ldots, x_n\}$, where $x_i \in \mathcal{X} = \mathbf{R}^d$.
2.Select appropriate kernel $K(x, x')$, C and ε, then construct and solve the problem

$$\begin{aligned} \min_{M,\delta,u} \quad & \frac{1}{2}\delta \\ \text{s.t.} \quad & \begin{pmatrix} (K \circ M + M + \frac{1}{C}I) & (u+e) \\ (u+e)^T & \delta \end{pmatrix} \succeq 0 \\ & -\varepsilon e \leq Me \leq \varepsilon e \\ & M \succeq 0, diag(M) = e \\ & u \geq 0 \end{aligned} \tag{4}$$

Get the optimal solution M^*, δ^* and u^* with SeDuMi.
3.Construct label $y^* = sgn(t_1)$, where t_1 is eigenvector corresponding to the maximal eigenvalue of M^*.

It is easy to extend the unsupervised classification algorithm to semi-supervised classification algorithm. For semi-supervised SVMs training, we can assume $(x_1, y_1), \ldots, (x_n, y_n)$ have labelled by experts and $x_{n+1}, \ldots, x_{n+N}$ are not labelled. Only adding the constraints $M_{ij} = y_i y_j, i, j = 1, 2, \ldots, n$ to the problem (4)will obtain the Semi-Supervised Classification Algorithm.

4 Experimental Results

4.1 Results of Unsupervised Classification Algorithm

In order to evaluate the performance of unsupervised classification algorithm, we will compared our unsupervised classification algorithm (L-SDP)with (ν-SDP)[11] and maximum margin clustering algorithm (C-SDP)[4]. Firstly we consider four synthetic data sets including data set AI, Gaussian, circles and joined-circles, which every data set has sixty points. $\varepsilon = 2$, $C = 100$ and Gaussian kernel with appropriate parameter $\sigma = 1$ are selected. Results are showed in Table 1. The number is the misclassification percent. From Table 1 we can find that the result of L-SDP is better than that of C-SDP and ν-SDP, moreover the time consumed are showed in Table 2. The numbers are seconds of CPU. From Table 2

Table 1. Classification results about three algorithms on four synthetic data sets

Algorithm	AI	Gaussian	circles	joined-circles
L-SDP	9.84	0	0	8.19
C-SDP	9.84	1.67	11.67	28.33
ν-SDP	9.84	1.67	1.67	11.48

Table 2. Computation time about three algorithms on four synthetic data sets

Algorithm	AI	Gaussian	circles	joined-circles
L-SDP	1425	1328	1087.6	1261.8
C-SDP	2408.9	1954.9	2080.2	2284.8
ν-SDP	2621.8	1891	1837.1	2017.2

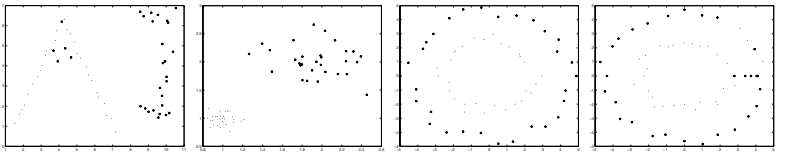

Fig. 1. Results by our unsupervised classification algorithm based on LSVMs on the four synthetic data sets including data set AI, Gaussian, circles and joined-circles

we can find that the speed of L-SDP is faster than those of C-SDP and ν-SDP, moreover almost half of the time consumed of others.

We also conduct our algorithm on the real data sets which can be obtained from http://www.cs.toronto.edu/ roweis/data.html, including Face and Digits data sets. As same to synthetic data sets, with thirty samples of every class of data sets. To evaluate clustering performance, a labelled data set was taken and the labels are removed , then run clustering algorithms, and labelled each of the resulting clusters with the majority class according to the original training labels, then measured the number of misclassification.The results are showed in Table 3 and the number is the misclassification percent. From Table 3 we

Table 3. Results about three algorithms on Face and Digits data sets

Algorithm	Digits32	Digits65	Digits71	Digits90	Face12	Face34	Face56	Face78
L-SDP	0	0	0	0	8.33	0	0	0
C-SDP	0	0	0	0	1.67	0	0	0
ν-SDP	0	0	0	0	1.67	0	0	0

can find that the result of L-SDP is almost same to C-SDP and ν-SDP except data set face12, but the time consumed are showed in Table 4. The numbers are seconds of CPU.

From Table 4 we can find that the speed of L-SDP is much faster than those of C-SDP and ν-SDP, moreover quarter of the time consumed of others at least.

4.2 Results of Semi-supervised Classification Algorithm

We test our algorithm to semi-supervised learning on the real data sets as same to section of unsupervised Classification Algorithm. As same to unsupervised classification algorithm, in order to evaluate the performance of semi-supervised

Table 4. Computation time about three algorithms on Face and Digits data sets

Algorithm	Digits32	Digits65	Digits71	Digits90	Face12	Face34	Face56	Face78
L-SDP	445.1	446.2	446.4	446.5	519.8	446.3	446.1	446
C-SDP	1951.8	1950.7	1951.6	1953.4	1954.5	1952.1	1950.3	1951
ν-SDP	1721.6	1721.5	1722.2	1722.8	1721.4	1722.1	1721	1719.7

Fig. 2. Every row shows a random sampling of images from a data set, the first ten images are in one class, while the rest ten images are in another class by L-SDP.

classification algorithm, we will compared our semi-supervised classification algorithm (Semi-L-SDP)with (Semi-ν-SDP)[11] and maximum margin clustering algorithm (Semi-C-SDP)[4]. We separate the data into labelled and unlabelled parts, and get rid of the labels of the unlabelled portion, then run semi-supervised classification algorithms to reclassify the unlabelled examples in use of the learning results, eventually measured the misclassification error on the original labels. Thirty samples of every class of data sets will be used. The results will be showed in Table 5 and the number is the misclassification percent. From Table 5 we can find that the result of Semi-L-SDP is almost same to Semi-ν-SDP and better than Semi-C-SDP except data set face12, but time consumed of CPU is much less than those of others. Results are showed in Table 6. The numbers are seconds of CPU. From Table 6 we can find that the speed of Semi-L-SDP is much

Table 5. Results about three algorithms on Face and Digits data sets

Algorithm	Digits32	Digits65	Digits71	Digits90	Face12	Face34	Face56	Face78
Semi-L-SDP	5	5	5	5	11.67	5	5	5
Semi-C-SDP	25	28.3	28.3	28.3	16.67	28.3	28.3	28.3
Semi-ν-SDP	5	5	5	5	3.3	5	5	5

Table 6. Computation time about three algorithms on Face and Digits data sets

Algorithm	Digits32	Digits65	Digits71	Digits90	Face12	Face34	Face56	Face78
Semi-L-SDP	606.8	607.8	607.9	608.3	653.9	608.4	608	608.2
Semi-C-SDP	1034	1036	1035.6	1035.8	1094.7	1033.1	1035.1	1035.8
Semi-ν-SDP	734.8	735.5	735.7	735.9	810.6	734.4	735.5	736

faster than those of Semi-C-SDP and Semi-ν-SDP, moreover almost half of the time consumed of others.

5 Conclusion

We have proposed efficient algorithms for unsupervised and semi-supervised classification problems based on Semi-definite Programming. From Section of experimental results we can learn that unsupervised and semi-supervised classification algorithms based on Lagrangian Support Vector Machines is much faster than other methods based on Bounded C-Support Vector Machines and Bounded ν-Support Vector Machines, and classification results are better than them.

In the future we will continue to estimate the approximation of SDP relaxation and get an approximation ratio of the worst case.

References

1. J.A.Hartigan, *Clustering Algorithms*, John Wiley and Sons, 1975.
2. B.Schoelkopf and A.Smola, *Learning with kernels: Support Vector Machines, Regularization, Optimization,and Beyond*, MIT Press, 2002.
3. G.Lanckriet, N.Cristianini, P.Bartlett, L.Ghaoui and M.Jordan, Learning the kernel matrix with semidefinite programming,*Journal of Machine learning research*, 5, 2004.
4. L.Xu, J.Neufeld, B.Larson and D.Schuurmans, Maximum margin clustering, *Advances in Neural Information Processing Systems 17(NIPS-04)*, 2004.
5. T.De Bie and N.Crisrianini, Convex methods for transduction, *Advances in Neural Information Processing Systems 16(NIPS-03)*, 2003.
6. N.Y.Deng and Y.J.Tian, *A New Method of Data Mining: Support Vector Machines*, Science Press, 2004.
7. T.Friess, C.N.Christianini, C.Campbell, The kernel adatron algorithm: a fast and simple learning procedure for support vector machines, *Proceeding of 15th Intl. Con Machine Learning*, Morgan Kaufman Publishers, 1998.
8. O.L.Mangasarian and D.R.Musicant, Successive overrelaxation for support vector machines, *IEEE Trans. Neural Networks*, 5,1999(10),1032-1037.
9. Nello Cristianini and John Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, 2000.
10. Jos F.Sturm, Using SeDuMi1.02, A Matlab Toolbox for Optimization over Symmetric Cones, *Optimization Methods and Software*, 11-12, 1999, 625-653.
11. Kun Zhao, Ying-jie Tian and Nai-yang Deng, Unsupervised and Semi-supervised Two-class Support Vector Machines, to appear.
12. O.L.Mangasarian and David R.Musicant,Lagrangian Support Vector Machines, *Journal of Machine Learning Research*,1,2001,161-177.