AMO - Advanced Modeling and Optimization, Volume 19, Number 1, 2017

Linear approach for twin-Hypersphere support vector machine

Saeed Ketabchi[a,*], Hossein Moosaei[b], Mohamad Razzaghi[a]

[a]*Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran*
[b]*Department of Mathematics, Faculty of science, University of Bojnord, Bojnord, Iran*

Abstract

Data classification by support vector Hypersphere which is a competitive method to all other methods usually ends in non-convex problems. In this paper, we will present a new method for classifying data based on the use of separating Hypersphere. Our main idea is to linearize the constraints of the problem to get rid of the non-convexity.

Keywords: Data Classification, Support Vector Machine, Hypersphere Classification, Nonlinear Programming.

1. Introduction

Classifying the data has been a subject of considerable attention in recent years. Among its applications are identification of the numbers and letters, voice recognition, face and handwriting, illness diagnosis, etc [2], [1] and [8]. Support vector machine (SVM) was first introduced to solve the pattern classification and regression problems by Vapnik and his colleagues. Many methods are proposed like separation through separating hyper plane with the maximum margin for binary classification, twin support vector machine, twin Hypersphere classification, etc.Twin support vector machine (TSVM) suggested in [6]. In this method we find two non-parallel hyperplane for binary classification. In [12] proposed a TWSVM classifier, termed the twin-hypersphere support vector machine (THSVM), for binary pattern recognition.The THSVM aims at generating two hyperspheres in the feature space such that each hypersphere contains as many as possible samples in one class and is as far as possible from the other one. Similar to the TWSVM, in THWSVM, we solve a pair of quadratic programming problems (QPPs), whereas, in SVMs, we solve a single QPP. In SVMs, the QPP has all data points in the constraints, but, in TWSVMs, they are distributed in the sense that patterns of one class give the constraints of the other QPP and vice versa. This strategy of solving two smaller sized QPPs, rather than one large QPP, makes THWSVMs work faster than standard SVMs. We know that the QPPs problems are non convex. In this paper, we change the non convex QPPs problems to convex QPPs

*corresponding author

Email addresses: sketabchi@guilan.ac.ir (Saeed Ketabchi), moosaei@ub.ac.ir (Hossein Moosaei), hmoosaei@gmail.com (Hossein Moosaei), razzaghi.mohamad@gmail.com (Mohamad Razzaghi)

problems and then solve this problems. The results shown that, our proposed method is efficient.
The signs and symbols used in this paper are as follows:
Let $a = [a_i]$ be a vector in R^n. By a_+ we mean a vector in R^n whose ith entry is 0 if $a_i < 0$ and equals a_i if $a_i \geq 0$. If f is a real valued function defined on the n-dimensional real space R^n, the gradient of f at x is denoted by $\nabla f(x)$ which is a column vector in R^n, and the $n \times n$ Hessian matrix of second partial derivatives of f at x is denoted by $\nabla^2 f(x)$. By A^T we mean the transpose of matrix A, and $\nabla f(x_0)^T d$ is called directional derivative of f at x_0 in direction d. For the two vectors x and y in the $n-$dimensional real space, $x^T y$ denotes the scalar product. For $x \in R^n$, $\|x\|$ denotes $2-$norm. A column vector of ones of arbitrary dimension will be indicated by e. For $A \in R^{m\times n}$ and $B \in R^{n\times l}$; the kernel $K(A;B)$ is an arbitrary function which maps $R^{m\times n} \times R^{n\times l}$ into $R^{m\times l}$. In particular, if x and y are column vectors in R^n then, $K(x^T;y)$ is a real number, $K(x^T;A^T)$ is a row vector in R^m , and $K(A;A^T)$ is an $m \times m$ matrix. The convex hull of a set S has been shown by $co\{S\}$. The identity $m \times n$ matrix will be denoted by $I_{m\times n}$.

2. Support Vector Machine and Twin Support Vector Machine

The issue of classifying data through support vector machine by considering hypothetical data $\{(x_1, y_1), ..., (x_n, y_n)\}$ where $y_i = \{\pm 1\}$ for $i = 1, ..., n$ is put into the following formula:

$$\begin{aligned} &\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \\ &s.t.\ y_i \left(w^T x_i + b\right) \geq 1 - \xi_i, \\ &\quad \xi_i \geq 0, i = 1, ..., n. \end{aligned} \tag{1}$$

Where C is the penalty parameter for data control. The main idea of the problem was first proposed by vapnik in 1963; while Bennet used a geometrical approach through convex shell to show that the dual problem is the following problem for which a very beautiful interpretation through the reduced convex shell is proposed [5].

$$\begin{aligned} &\min_{u,v} \frac{1}{2} \|A^T u - B^T v\|^2 \\ &s.t.\ e^T u = 1,\ e^T v = 1 \\ &\quad 0 \leq u \leq De, 0 \leq v \leq De. \end{aligned} \tag{2}$$

Where A, B are the data available in two classes $1, -1$ and $D < 1$ is the reduced convex shell parameter. By solving (1) and determining vector w and scalar b, the separating hyperplane $w^T x + b = 0$ for data classification is achieved [4], [3].
To solve the SVM problem in many papers, here dual problem is solved and the solution to the initial problem is achieved from the solution gained for dual problem by writing KKT conditions, regardless of the fact the initial problem might be well less than the dual problem in terms of variables, or the number of constraints in dual problem is less than the initial

problem.
Rather than finding the separating hyperplane for the desired data, we will find 2 non-parallel hyperplane by solving two problems of convex planning and the distance between data and hyperplane is used as a criterion for classification in this method. The issue of programming the twin support vector machine is put into the following formula:

$$\begin{array}{ll} \min & \frac{1}{2}\left\|Aw^{(1)}+eb^{(1)}\right\|^2+c_1e^T\xi^{(1)} \\ s.t. & -\left(Bw^{(1)}+eb^{(1)}\right)\geq e-\xi^{(1)}, \\ & \xi^{(1)}\geq 0_e. \end{array} \tag{3}$$

$$\begin{array}{ll} \min & \frac{1}{2}\left\|Bw^{(2)}+eb^{(2)}\right\|^2+c_2e^T\xi^{(2)} \\ s.t. & \left(Aw^{(2)}+eb^{(2)}\right)\geq e-\xi^{(2)}, \\ & \xi^{(2)}\geq 0_e. \end{array} \tag{4}$$

By solving the 2 above problems, hyper panels $x^Tw^{(1)}+b^{(1)}=0$, $x^Tw^{(2)}+b^{(2)}=0$ will be achieved for separating the hypothetical data. For separating more than 2 classes, TSVM problem will be rewritten as follows:

$$\begin{array}{ll} \min & \frac{1}{2}\left\|Aw^{(1)}+eb^{(1)}\right\|^2+c_1{e_2}^T\xi+c_2e_3^T\eta \\ s.t. & -\left(Bw^{(1)}+e_2b^{(1)}\right)\geq e-\xi, \\ & -\left(Cw^{(1)}+e_3b^{(1)}\right)\geq e_3(1-\epsilon)-\eta, \\ & \xi\geq 0_e,\ \eta\geq 0_e. \end{array} \tag{5}$$

$$\begin{array}{ll} \min & \frac{1}{2}\left\|Bw^{(2)}+e_2b^{(2)}\right\|^2+c_3{e_1}^T\xi^*+c_4e_3^T\eta^* \\ s.t. & \left(Aw^{(2)}+e_1b^{(2)}\right)\geq e_1-\xi^*, \\ & \left(Cw^{(2)}+e_3b^{(2)}\right)\geq e_3(1-\epsilon)-\eta^*, \\ & \xi^*\geq 0_e,\ \eta^*\geq 0_e. \end{array} \tag{6}$$

Like SVM problem, problems (5) and (6) are convex problems and the answer to the initial and dual problems are equal to one another and solving the dual problem will lead us to the initial problems answer. The dual problems of problems (5) and (6) are written as follows.

$$\begin{array}{ll} \max & -\frac{1}{2}\gamma^TN(H^TH)^{-1}V^T\gamma+e_4^T\gamma \\ s.t. & 0\leq\gamma\leq F. \end{array} \tag{7}$$

$$\begin{array}{ll} \max & -\frac{1}{2}\rho^TP(G^TG)^{-1}P^T\rho+e_5^T\rho \\ s.t. & 0\leq\rho\leq F^*. \end{array} \tag{8}$$

Where $H=[A\ \ e_1]$, $G=[B\ \ e_2]$, $M=[C\ \ e_3]$, $N=[G;M]$, $e_4=[e_2;e_3(1-\epsilon)]$, $e_5=[e_1;e_3(1-\epsilon)]$, $\gamma=[\alpha;\beta]$, $F=[c_1e_2;c_2e_3]$, $P=[H;M]$, $F^*=[c_3e_1;c_4e_3]$.

3. Hypersphere and Twin Hypersphere

One of the ideas proposed for data classification in recent years is the classifying data by support vector hyperspheres. In this approach, we seek to find a hypersphere with the shortest radius which covers all data. The modeling for separating hypersphere is as follows [10]:

$$\begin{aligned} &\min \ R^2 \\ &s.t. \ \|x_i - c\|^2 \leq R^2, \\ &\forall i = 1, ..., n. \end{aligned} \tag{9}$$

Where x_i represents the hypothetical data, R represents the radius, and C presents the center of super sphere. Generally, the DATA might have some form of distribution, thus a hypersphere which can cover all the data might seem not affordable. To solve this problem, the data which are far from other data are put inside the hypersphere through an error parameter. This error parameter is chosen in such a way that the whole data minimizes error. Modeling in this mode is turned into the following model:

$$\begin{aligned} &\min_{R,\, c,\, \xi} \ R^2 + C \sum_{i=1}^{n} \xi_i \\ &s.t. \|x_i - c\|^2 \leq R^2 + \xi_i, \\ &\xi_i \geq 0, \forall i = 1, ..., n. \end{aligned} \tag{10}$$

As we know, problem (10) is non-convex, thus we can conclude from the constraints of the first problem:

$$Hessian = \begin{pmatrix} 2I & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Which is infinite. In [13], without paying attention to the non-convexity of the problem, the s to the solution first problem is calculated by going to the dual problem and solving it. Based on the sufficient and necessary condition, KKT is incorrect. Using the idea of twin support vector machine stated previously, the problem of finding a hypersphere which stores the data of the first set in herself and the data of the second set outside itself and vice versa is modeled as follows which is named twin support vector hypersphere:

$$\begin{aligned} &\min \frac{1}{2} \sum_{i \in I_1} \|x_i - c_1\|^2 - v_1 R_1^2 + C_1 \sum_{j \in I_2} \xi_j \\ &s.t. \ \ \|x_j - c_1\|^2 \geq R_1 - \xi_j, \\ &\xi_j \geq 0, \ j \in I_2. \end{aligned} \tag{11}$$

$$\begin{aligned}
&\min \frac{1}{2}\sum_{i\in I_2}\|x_i-c_2\|^2-v_2R_2^2+C_2\sum_{j\in I_1}\xi_i\\
&s.t.\ \|x_i-c_2\|^2\geq R_2-\xi_i,\\
&\quad\ \xi_i\geq 0,\ i\in I_1.
\end{aligned}\tag{12}$$

This model finds two hyperspheres, one for each class, and classifiers points according to which hyper sphere a given point is closest to. Noting the goal and Hessian function of the problems, we will clearly realize that the problems mentioned above are not convex. In many papers including [12] and [13], without considering the non-convexity of first problem, we go to the dual problem and write KKT conditions and the answer to the first problem is drawn mistakenly from the answer to the dual problem. To solve these problems, an effective method will be proposed later. To solve the problems just outlined, we linearize the provisions. Ultimately, the problems will be transformed into the following form:

$$\begin{aligned}
&\min \frac{1}{2}\sum_{i\in I_1}\|x_i-c_1\|^2-v_1R_1+C_1\sum_{j\in I_2}\xi_j\\
&s.t.\ \|x_j\|^2-2x_j^Tc_1\geq R_1-\xi_j,\\
&\quad\ \xi_j\geq 0,\ j\in I_2.
\end{aligned}\tag{13}$$

$$\begin{aligned}
&\min \frac{1}{2}\sum_{i\in I_2}\|x_i-c_2\|^2-v_2R_2+C_2\sum_{j\in I_1}\xi_i\\
&s.t.\ \|x_i\|^2-2x_i^Tc_2\geq R_2-\xi_i,\\
&\quad\ \xi_i\geq 0,\ i\in I_1.
\end{aligned}\tag{14}$$

We may conclude from [11] that these problems are problems of convex quadratic planning with linear provisions and we use $Quadprog$ toolbox in $Matlab$ software to solve them. In the next section, we will present the numerical results in a table and study the effectiveness of this method.

4. Numerical Experiment

In this section, we will study the accuracy and speed of the method proposed through the data extracted from UCI. For calculations, we used Matlab 2014a and a computer with a $Corei5$ processor and 8 gigabytes ram. To measure the precision of the method stated, we randomly selected 20% of the data and applied this action 5 times to the whole data and calculated their mean as a percentage of precision.
In the first problem, we determine randomly some arbitrary points in two classes of A and B which are approximately separated from each other linearly based on given MATLAB code 1 in the following (here, we created 150 points for class A and 100 points for class B). These data are produced randomly within the interval $[-50, 50]$.The accuracy rate of separating in

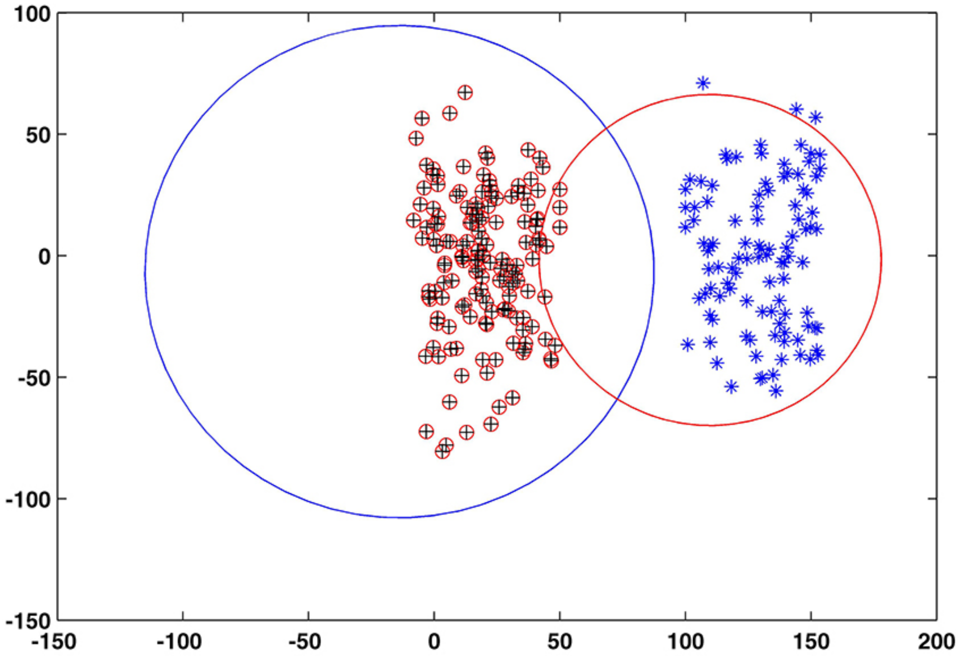

Figure 1: Classification results of linear hypersphere on generated dataset.

this problem is 100%.In Fig 1, the given separating hyperspheres has been shown with red and blue color.

MATLAB code 1

```
pl=inline('(abs(x)+x)/2');tic;
M=rand(m1,n);
M=100*(M-0.5*spones(M));
M(:,2)=M(:,1)+1*ones(m1,1)+100*rand(m1,1)+100*rand(m1,1);
N=rand(m2,n);
N=100*(N-0.5*spones(N));
N(:,2)=N(:,1)-1*ones(m2,1)-100*rand(m2,1)-100*rand(m2,1);
uu=5*rand(3,n);
uu1=uu;uu1(:,2)= uu1(:,1)+1*ones(3,1);
uu2=uu;uu2(:,2)= uu2(:,1)-1*ones(3,1);
M=[M;uu1;10 0]; N=[N;uu2;30 -20];m1=m1+4;m2=m2+4;m=m1+m2;
xM=[-50:40*rand:  50];yM=xM+1;xN=[-50:20*rand:50];yN=xN-1;
plot(M(:,1),M(:,2),'oblack',N(:,1),N(:,2),'*bl');
axis square
format short ;[m1 m2 n toc],[max(M(:,1)) min(N(:,1))]
```

To further test the performance of linearized Hypersphere, we run this algorithm on several UCI benchmark data sets. The following table represents the numerical results achieved.

Tab. 1. Hypersphere classification on some benchmark data sets.

Datas	Size	Accuracy (%)	Time (s)
Haberman	306×3	73.53	0.52
Spect	237×22	58.8	0.51
German	1000×24	70	1.21
housevotes84	435 ×16	89.42	0.54
Diagnosis	100×9	85	0.44
ionosphere	351×34	66.95	0.69

In this paper we present a new idea for solving the hypersphere classification problem. We convert the non convex quadratic minimization problem into an Linear constrained problem. The above table is indicative of effectiveness and high speed of calculations in solving library problems.

References

[1] Joachims T, Text categorization with support vector machines: learning with many relevant features,European conference on machine learning no. 10[C] 1398. Chemnitz, Springer, pp 137-142.

[2] Cao L, Tay FEH financial forecasting using support vector machines, Neural Comput Appl 10(2),184-192.

[3] Vapnik, V. N., Chervonenkis, A. Theory of pattern recognition. Moscow: Nauka [in Russian], 1974.

[4] Vapnik VN Statistical learning theory. Wiley, New York, 1998.

[5] Kristin P. Bennett , Erin J. Bredensteiner, Duality and Geometry in SVM Classifiers. In Proc. 17th International Conf. on Machine Learning,page 57–64, 2000.

[6] Xu, Yitian, A Guo, Rui, A Wang, Laisheng , A Twin Multi-Class Classification Support Vector Machine, Cognitive Computation, Volume 5, Pages 580-588.

[7] P.M. Murphy, D.W.Aha, UCI repository of machine learning database, 1992.

[8] Osuna E, Freund R, Girosi F, Training support vector machines: an application to face detection, roceedings of the 1997 conference computer vision and pattern recognition[C]. IEEE Computer Society,Washington, pp 130-136.

[9] Yuan-Hai Shao, Zhen Wang, Wei-Jie Chen, Nai-Yang Deng, A regularization for the projection twin support vector machine, Knowledge-Based Systems, Volume 37, January 2013, Pages 203-210.

[10] David M. J. Tax and Robert P. W. Duin. 2004. Support Vector Data Description. Mach. Learn. 54, 1 (January 2004), 45-66.

[11] Mokhtar S. Bazaraa, Hanif D. Sherali, C. M. Shetty. NONLINEAR PROGRAMMING Theory and Algorithms, 3rd ed 2006.

[12] Peng, Xinjun and Xu, Dong, Twin support vector hypersphere (TSVH) classifier for pattern recognition, Neural Computing and Applications, , Volume 24, 2014, Pages 1207-1220.

[13] Xinjun Peng, Dong Xu, A twin-hypersphere support vector machine classifier and the fast learning algorithm, Information Sciences, 2013, Pages 12-27.