A New Formulation for Classification by Ellipsoids

Ayşegül Uçar[1], Yakup Demir[1], and Cüneyt Güzeliş[2]

[1] Electrical and Electronics Engineering Department, Engineering Faculty, Fırat University, Elazig, Turkey
agulucar@firat.edu.tr,ydemir@firat.edu.tr
[2] Electrical and Electronics Engineering Department, Dokuz Eylül University, Kaynaklar Campus, İzmir, Turkey
guzelis@eee.deu.edu.tr

Abstract. We propose a new formulation for the optimal separation problems. This robust formulation is based on finding the minimum volume ellipsoid covering the points belong to the class. Idea is to separate by ellipsoids in the input space without mapping data to a high dimensional feature space unlike Support Vector Machines. Thus the distance order in the input space is preserved. Hopfield Neural Network is described for solving the optimization problem. The benchmark Iris data is given to evaluate the formulation.

1 Introduction

Many methods for binary classification were proposed. Support Vector Machines (SVMs) are one of the optimization-based approaches for solving supervised machine learning problems [1-2]. The basic ability of SVM is to implicitly transform data to a high dimensional space and to construct a linear binary classifier without having to perform any computations in the high dimensional space. Hence SVM doesn't suffer from problems such as the dimensionality of the data and the sparsity of data in the high-dimensional space. The hyperplanes in the high dimensional transform space result in complex decision surfaces in the input data space. However the SVM classifiers can not be preserve the distance in input space since they separate data in the feature space [3].

In this paper, the classification in the input space is aimed. An ellipsoid is used as main tool. Few authors attempted to solve the pattern separation problem by ellipsoids. Among these, Barnes, Boyd and Astorino presented valid the formulations and algorithms for especially ellipsoidal separable problem [4-7]. On the other hand, Frigue developed semidefinite programming formulations for both ellipsoidal separable and non-ellipsoidal separable problems. However a robust classification remains still a need for pattern separation area. We give a new formulation by underlying both structural error and empirical error in SVM classifiers in this work.

The remainder of this paper is organized as follows. Section II reviews the SVM classifiers and their some limitations. The proposed formulation and solution algorithm are shown in Section III. In Section IV, the performance of the method is illustrated on Iris data. Conclusions are summarized in Section V.

F.A. Savacı (Ed.): TAINN 2005, LNAI 3949, pp. 100 – 106, 2006.

2 Reviews to SVM Classifier

The SVM learns the decision function that maximally separates two classes of training samples by the prime optimization problem [1]:

$$\begin{aligned} \min_{w,b,\xi} \quad & J(\vec{w},\vec{\xi}) = \frac{1}{2} w^T w + C\sum_{i=1}^{l} \xi_i \\ st. \quad & y_i\left[w^T \varphi(x_i) + b\right] \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1,...,l \end{aligned} \tag{1}$$

where $\varphi(\mathbf{x})$ is a nonlinear transformation vector from the input space to the feature space, b is a bias, w is an adjustable weight vector, and C>0 is a tradeoff constant between training error and margin for slack variables $\vec{\xi}$ allowing to the misclassification of training sample. The optimization problem is solved constructing the following dual quadratic programming:

$$\begin{aligned} \min_{\alpha} \quad & J_D(\alpha) = -\frac{1}{2}\sum_{i,j=1}^{l} y_i y_j K(x_i, x_j)\alpha_i \alpha_j + C\sum_{i=1}^{l} \alpha_i \\ st. \quad & \sum_{i=1}^{1} \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C, i = 1,...,l \end{aligned} \tag{2}$$

where the kernel function is expressed by $K(x_i, x_j) = \varphi^T(x_i)\varphi(x_j)$. α_i are Lagrange multipliers.

2.1 Some Limitations of SVM Classifiers

SVM classifiers have some limitations. Firstly, there is a question whether the feature space distances describe meaningful structures of the input space data. In this section, this issue is investigated for Radial Basis Function (RBF) kernels and polynomial kernels. If the distance is preserved in both spaces, then the distance in the feature space is given as a function of the distance in the input space.

In case of RBF kernel, $k(x_i, x_j) = \exp\left(-\left\|x_i - x_j\right\|^2 / 2\sigma^2\right)$, the distance in feature space is given as:

$$d(\Phi(x_i), \Phi(x_j)) = \sqrt{2 - 2\exp\left(-\frac{d(x_i, x_j)^2}{2\sigma^2}\right)}. \tag{3}$$

This implies that the distance between two observations in a RBF feature space is bounded by $\lim_{d(x_i,x_j)\to\infty} d\left(\Phi(x_i),\Phi(x_j)\right)=\sqrt{2}$ and σ regulates the sensitivity regarding their input space distance. Hence as the distance increases, classification results in error since they approximate to each as seen by Fig. 1(a) [8-10].

On the other hand, in the polynomial kernels, $k\left(x_i,x_j\right)=\left(\left\langle x_i,x_j\right\rangle+\theta\right)^d$, the distance in the feature space depends on the absolute position in input space. In the standardized polynomial kernels, $k\left(x_i,x_j\right)=\left(\left\langle x_i,x_j\right\rangle\Big/\sqrt{\left\langle x_i,x_j\right\rangle\left\langle x_i,x_j\right\rangle}+\theta\right)^d$, the feature space distance is a function of the angle α between x_i and x_j in the input space

$$d\left(\Phi(x_i),\Phi(x_j)\right)=\sqrt{2(1+\theta)^d-2(\cos\alpha+\theta)^d}\ . \tag{4}$$

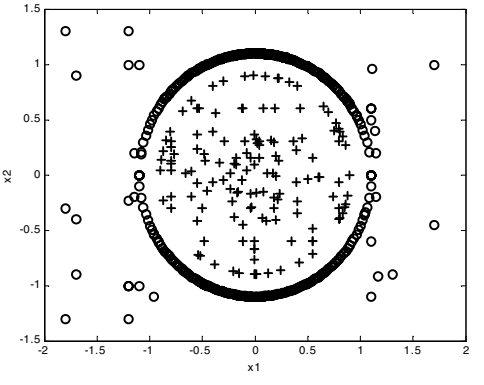

Fig. 1. An artificial data constructed for example

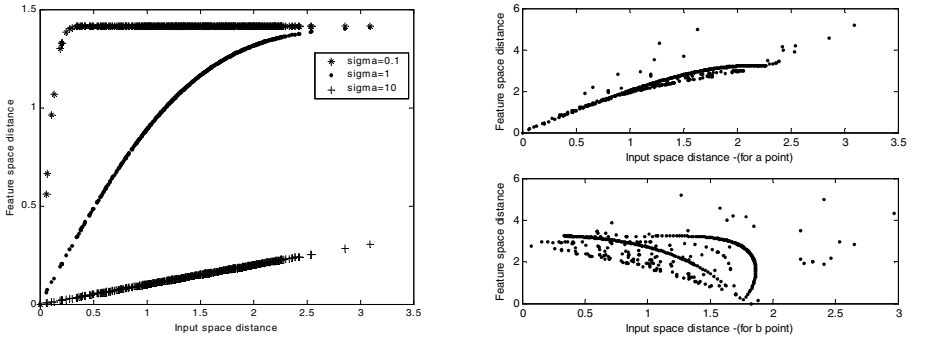

Fig. 2. Distance order input vs. feature space. (a) RBF kernel. (b) Polynomial kernel.

In order to show whether the distance preserve, we generated an artificial data of two-class including 494 data in Fig. 1(b). Choosing two reference points, a=[-1.1500-0.2000] and b=[0.5 0.5798], we computed the distances in input space and feature space to this points. In addition we enquired whether the distance order is preserved in the feature space. As seen from the distance to a (b) points tabulated in Table 1 (2) and the Euclidean distances to a (b) points illustrated in Fig. 2, the distance order in the input space is not preserved for a polynomial kernel. In this paper, we present a new formulation in the input space to get rid of this disadvantage. SVM has also the other limitations in that an analytical/structural point of view, the details can be found in [11].

Table 1. The distances to a point

x1	x2	Input space	Feature space
-1.2000	-1.0000	2.3207	3.7795
1.7000	1.0000	1.2714	3.8236
-1.7000	-0.4000	2.4083	4.3480
-1.8000	1.3000	2.4101	6.0186
-1.8000	-1.3000	2.9705	6.0684

Table 2. The distances to b point

x1	x2	Input space	Feature space
0.6500	0.6225	1.9790	2.9706
0.8600	-0.2653	2.0111	2.9748
0.4600	0.9992	2.0075	3.2122
1.1500	-0.2000	2.3000	3.3171
0.8800	-0.1887	2.0300	2.9759

3 Proposed formulation

A new optimization formulation for a robust classification is proposed by using ellipsoid in this study. In the proposed formulation, each class assigned with data is represented by an ellipsoid. It is aimed that the ellipse includes the data belongs to the class and the other to be excludes. For $c \in \Re^n$ and $M \in \Re^{nxn}$, the separating ellipsoid is defined as

$$\varepsilon_{M,c} = \left\{ x \in \Re^n \middle| (x-c)^T M (x-c) \leq 1 \right\}; \tag{5}$$

where c is the center of the ellipsoid and M is a symmetric positive matrix that determines its shape.

If the problem is separable, the empirical error equal to zero and the structural error is defined as 1/det(M) similar to SVM classifier. If the problem is non-separable, then misclassified data must be penalized. In this case, the empirical error is not equal to zero [8],[12]. Hence the proposed formulation is written as

$$\min_{c,M} \mathrm{E} = \frac{B}{2}\sum_{i=1}^{L} g\left(y_i.\left((x_i-c)^T M\left(x_i-c\right)-1\right)\right)+\frac{A}{2}.\frac{1}{|M|}, \tag{6}$$

where A and B parameters give the tradeoff between the volume of the optimal separation and error. An appropriate selection of these parameters is very important to minimize the objective function. Here the loss function and its derivative is chosen to be

$$\begin{aligned} g(\xi)&=\xi \quad \xi>0 & \qquad g'(\xi)&=1 \quad \xi>0 \\ g(\xi)&=0 \quad \xi\le 0 & \qquad g'(\xi)&=0 \quad \xi\le 0 \end{aligned}. \tag{7}$$

First term, if a data is outside of ellipsoid, a positive term adds to the objective function. Second term is used to find the minimum volume ellipsoid covering the data in the class. This term can be easily obtained taking into consideration the geometric interpretation of the problem. While the algorithm carries on minimization 1/(det(M), the data number excluding of ellipsoid tries to minimize in same time. In other words, even when data is not perfectly separated by ellipsoids, our formulation can find an optimal solution.

3.1 Hopfield Network Approach

To solve the optimization problem, we construct a Hopfield network [8]. The Hopfield neural network is fully connected continuous time network that employs units with analog input and output. The basis idea is to encode the objective function and the problem constraints in terms of an appropriate energy function, Lyapunov function that can be minimized by the network architecture. In this study, synchronously operating two Hopfield networks are constructed. The network weights to be estimated are centers, c and covariance matrixes, M of ellipsoids. The behavior of each Hopfield network is evaluated by

$$\frac{dU}{dt} = -\frac{\partial E}{\partial v}, \tag{8}$$

$$v = f(U), \tag{9}$$

where E is the energy function, U is the input neural units, and v is the output of neural units. In this work, f is chosen as linear activation function for both networks. The input of each neural unit is updated by

$$U_{new} = U_{old} - \eta\frac{\partial E}{\partial v}, \tag{10}$$

where η represents learning rate.

The dynamics of the network is given according to the following differential equations:

$$\frac{\partial U_c}{\partial t} = B\sum_{i=1}^{L} y_i M(x_i - c) g'\left(y_i\left((x_i - c)^T M(x_i - c) - 1\right)\right) \tag{11}$$

$$\frac{\partial U_M}{\partial t} = -\frac{B}{2}\sum_{i=1}^{L} y_i (x_i - c)(x_i - c)^T g'\left(y_i\left((x_i - c)^T M(x_i - c) - 1\right)\right) + \frac{A}{2}\frac{\mathrm{M}^{-\mathrm{T}}}{|M|} \tag{12}$$

4 Examples

To see how the proposed formulation works we used Fisher's Iris data set. This data set contains 150 feature vectors of dimension 4, which belong to three classes representing different IRIS subspecies. Each class contains 50 feature vectors. One of the three classes is well separated from the other two, which are not easily separable due to the overlapping of their convex hulls. In the first example, we consider only two-class separable problem for two characters of the data set. In the second example, we carry out multi-class separation by reducing the problem separating classes to independent separation problems involving two classes each. We separate each class from one class consisting of the other two. We used the first 75 examples contained in the IRIS data set for the training the remaining for testing. Then we present comparatively results the classifications by SVM on the iris data.

In two-class separable example, we chosen as B=9, A=9 and learning rates: 0.08 and 0.0002 for c and M. respectively and operated algorithm for 1000 epoch. In multi-class example, we chosen as B=4, A=5 and learning rates: 0.08 and 0.0001 for c and M. respectively and operated algorithm for 4000 epoch. For SVM, we accepted as C=5000 as in [13]. We carried out the simulations in MATLAB.

For two-class separable problem, both SVM and our method result in % 100 accuracy in both test and training. On the other hand, in multi-class problem, our method yields % 94.667 accuracy in test and %94.667 in training, respectively while SVM yields %100 and %93.333 accuracy in test and training. However SVM can give

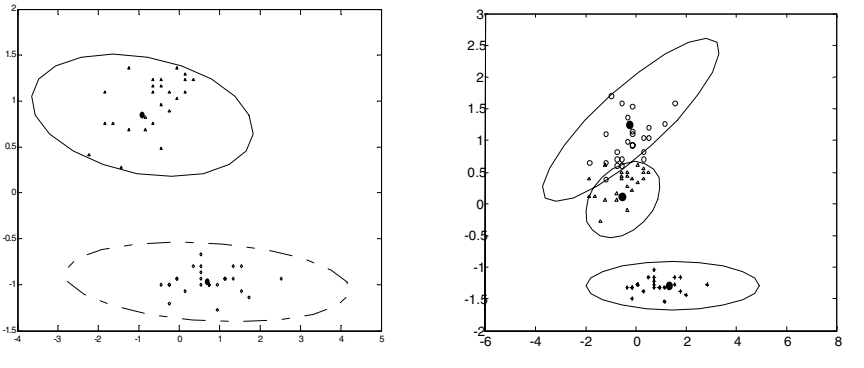

Fig. 3. Separating ellipsoids obtained using the proposed method. (a) Two-class problem. (b) Three-class problem.

better performance in terms of smaller margin. The performance of our method is illustrated in Fig. 3.

5 Conclusions

The SVM classifiers cannot preserve in the feature space the distance order in input space. In this paper we have investigated this limitation of the SVM classifier. This idea has activated us to obtain SVM like classifiers in input space. We have proposed a new optimization formulation for a robust classification in the input space. We have carried out the solution of the objective function including centers and covariance matrixes of separating ellipsoids by the Hopfield Neural Network. In particular, it is remarkable that this formulation allows us to robust classification by minimum volume ellipsoids in the input space.

References

1. Vapnik, V.: Statistical Learning Theory. John Wiley, New York (1998)
2. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines, Cambridge University Press (2000)
3. Zhang, B.: Is the Maximal Margin Hyperplane Special in a Feature Space? Hewlett-Packard Research Laboratories Palo Alto (2001)
4. Barnes, E.R.: An Algorithm for Separating Patterns by Ellipsoids, IBM. J. Res. Develop, Vol. 26. (1982) 759-764
5. Vandenberghe, L., Boyd S.: Applications of Semidefinite Programming. Technical Report of California University (1998)
6. Glineur, F.: Pattern Separation Via Ellipsoids and Conic Programming, Mémoire de D.E.A., Faculté Polytechnique de Mons, Mons, Belgium, September (1998)
7. Astorino, A. Gaudioso, M.: Ellipsoidal Separation for Classification Problems. Optimizations Methods and Software. Vol. 20. (2005) 267-276
8. Doğan, H.: Gradient Networks Design for Clustering in Novel Optimization Frameworks. Dokuz Eylül University, PhD. Thesis, December (2004)
9. Kruss, M.: Nonlinear Multivariate Analysis with Geodesic Kernels. Berlin Technical University, Thesis, February (2002)
10. Zhang, Z.: Learning Metrics Via Discriminant Kernels and Multidimensional Scaling: Toward Expected Euclidean Representation. ICML (2003) 872-879
11. Lyhyaoui, A., Martinez, M., Mora, I., Vaquez, M., Sancho, J.-L., Figueiras-Vidal, A.R.Sample Selection Via Clustering to Construct Support Vector-Like Classifiers. IEEE Trans. Neural Networks, Vol. 10. (1999) 1474-1481
12. Tax, D.M.J., Duin, R.P.W.: Support Vector Domain Description. Pattern Recognition Letters, Vol. 20. (1999) 1191-1199
13. Vincent, W.: SVM and Co-Training. Honc Konc Bapist University, Technical Report, (2002)