Pattern Classification via Single Spheres

Jigang Wang, Predrag Neskovic, and Leon N. Cooper*

Institute for Brain and Neural Systems,
Department of Physics,
Providence, RI 02912 USA
{jigang, pedja, Leon_Cooper}@brown.edu
http://physics.brown.edu/physics/researchpages/Ibns/index.html

Abstract. Previous sphere-based classification algorithms usually need a number of spheres in order to achieve good classification performance. In this paper, inspired by the support vector machines for classification and the support vector data description method, we present a new method for constructing single spheres that separate data with the maximum separation ratio. In contrast to previous methods that construct spheres in the input space, the new method constructs separating spheres in the feature space induced by the kernel. As a consequence, the new method is able to construct a single sphere in the feature space to separate patterns that would otherwise be inseparable when using a sphere in the input space. In addition, by adjusting the ratio of the radius of the sphere to the separation margin, it can provide a series of solutions ranging from spherical to linear decision boundaries, effectively encompassing both the support vector machines for classification and the support vector data description method. Experimental results show that the new method performs well on both artificial and real-world datasets.

1 Introduction

When objects are represented as d-dimensional vectors in some input space, classification amounts to partitioning the input space into different regions and assigning unseen objects in those regions into their corresponding classes. In the past, people have used a wide variety of shapes, including rectangles, spheres, and convex hulls, to partition the input space.

Spherical classifiers were first introduced into pattern classification by Cooper in 1962 and subsequently studied by many other researchers [1,2,3,4]. One well known classification algorithm consisting of spheres is the Restricted Coulomb Energy (RCE) network. The RCE network, first proposed by Reilly, Cooper, and Elbaum, is a supervised learning algorithm that learns pattern categories by representing each class as a set of prototype regions - usually spheres [5,6]. The RCE network incrementally creates spheres around training examples that are not covered, and it adaptively adjusts the sizes of spheres so that they do

* This work is partially supported by ARO under grant DAAD19-01-1-0754. Jigang Wang is supported by a dissertation fellowship from Brown University.

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 241–252, 2005.

not contain training examples from different classes. After the training process, only the set of class-specific spheres is retained and a new pattern is classified based on which sphere it falls into and the class affiliation of that sphere.

Another learning algorithm that is also based on spherical classifiers is the set covering machine (SCM) proposed by Marchand and Shawe-Taylor [7]. In their approach, the final classifier is a conjunction or disjunction of a set of spherical classifiers, where every spherical classifier dichotomizes the whole input space into two different classes with a sphere. The set covering machine, in its simplest form, aims to find a conjunction or disjunction of a minimum number of spherical classifiers such that it classifies the training examples perfectly.

Regardless of whether the influence of a sphere is local (as in the RCE network) or global (as in the SCM), classification algorithms that use spheres normally need a number of spheres in order to achieve good classification performance, and therefore have to deal with difficult theoretical and practical issues such as how many spheres are needed and how to determine the centers and radii of the spheres. In this paper, inspired by the support vector machines (SVMs) for classification [8,9,10] and the support vector data description (SVDD) method [11,12], we propose a new method, which computes a single sphere that separates data from different classes with the maximum separation ratio. In contrast to previous methods that construct spheres in the input space, the proposed method constructs the separating sphere in the feature space induced by the kernel. Because the class of spherical boundaries in the feature space actually represents a much larger class than in the input space, our method is able to construct a single sphere in the feature space that separates patterns that would otherwise be inseparable when using a sphere in the input space.

Furthermore, when the ratio of the radius of the separating sphere to the separation margin is small, a sphere is constructed that gives a compact description of one class, coinciding with the solution of the SVDD method; and when the ratio is large, the solution effectively coincides with the maximum margin hyperplane solution. Therefore, by adjusting the ratio, the new method effectively encompasses both the support vector machines for classification and the SVDD method for data description, and may lead to better generalization performance than both methods.

The remainder of the paper is organized as follows. In Section 2 we give a brief overview of the support vector data description method that computes a minimum enclosing sphere to describe a set of data from a single class. In Section 3, we propose our new algorithm, which extends the SVDD method by computing a single sphere that separates data from different classes with the maximum separation ratio. In Section 4 we test the new algorithm on both artificial and real-world datasets. Concluding remarks are given in Section 5.

2 Support Vector Data Description

The basic idea of the SVDD method is to construct a minimum bounding sphere to describe a set of given data. The minimum bounding sphere, which is defined as the smallest sphere enclosing all data, was first used by Schölkopf, Burges, and Vapnik to estimate the VC-dimension of support vector classifiers and later applied by Tax and Duin to data description [11,12].

Given a set of training data $x_1, \ldots, x_n \in \mathbb{R}^d$, the minimum bounding sphere S, characterized by its center c and radius R, can be found by solving the following constrained quadratic optimization problem

$$\min_{c,R} \; R^2 \;, \tag{1}$$

subject to the constraints

$$\|x_i - c\|^2 \le R^2 \quad \forall i = 1, \ldots, n \;. \tag{2}$$

To allow for the possibility of some examples falling outside of the sphere, one can relax the constraints (2) with a set of soft constraints:

$$\|x_i - c\|^2 \le R^2 + \xi_i \quad \forall i = 1, \ldots, n \;, \tag{3}$$

where $\xi_i \ge 0$ are slack variables introduced to allow some examples to have larger distances. To penalize large distances to the center of the sphere, one can therefore minimize the following quadratic objective function

$$\min_{c,R,\xi_i} \; R^2 + C \sum_{i=1}^{n} \xi_i \;, \tag{4}$$

under the constraints (3), where $C > 0$ is a constant that controls the trade-off between the size of the sphere and the number of examples that possibly fall outside of the sphere.

Using the Lagrange multiplier method, the constrained quadratic optimization problem can be formulated as the following Wolfe dual form

$$\min_{\alpha_i} \sum_{i,j} \alpha_i \alpha_j \langle x_i, x_j \rangle - \sum_i \alpha_i \langle x_i, x_i \rangle \tag{5}$$

subject to the constraints

$$\sum_{i=1}^{n} \alpha_i = 1 \quad \text{and} \quad 0 \le \alpha_i \le C \quad \forall i = 1, \ldots, n \;. \tag{6}$$

Solving the dual quadratic programming problem, one obtains the Lagrange multipliers α_i for all $i = 1, \ldots, n$, which give the center c of S as a linear combination of x_i

$$c = \sum_{i=1}^{n} \alpha_i x_i \;. \tag{7}$$

According to the Karush-Kuhn-Tucker (KKT) optimality conditions, we have

$$\begin{aligned} \alpha_i = 0 &\Rightarrow \|x_i - c\|^2 < R^2 \quad \text{and} \quad \xi_i = 0 \\ 0 < \alpha_i < C &\Rightarrow \|x_i - c\|^2 = R^2 \quad \text{and} \quad \xi_i = 0 \\ \alpha_i = C &\Rightarrow \|x_i - c\|^2 \ge R^2 \quad \text{and} \quad \xi_i \ge 0 \;. \end{aligned}$$

Therefore, only α_i that correspond to training examples x_i which lie either on or outside of the sphere are non-zero. All the remaining α_i are zero and the corresponding training examples are irrelevant to the final solution. Knowing c, one can subsequently determine the radius R from the KKT conditions by letting

$$R^2 = \langle x_i, x_i \rangle - 2\sum_{j=1}^{n} \alpha_i \langle x_i, x_j \rangle + \sum_{j,l} \alpha_j \alpha_l \langle x_j, x_l \rangle \tag{8}$$

for any i such that $0 < \alpha_i < C$.

In practice, training data of a class is rarely distributed spherically, even if the outermost examples are excluded. To allow for more flexible descriptions of a class, one can apply the kernel trick by replacing the inner products $\langle x_i, x_j \rangle$ in the dual problem with suitable kernel functions $k(x_i, x_j)$. As a consequence, training vectors x_i in $\mathbb{R}^d$ are implicitly mapped to feature vectors $\Phi(x_i)$ in some high dimensional feature space $\mathbb{F}$ such that inner products in $\mathbb{F}$ are defined as $\langle \Phi(x_i), \Phi(x_j) \rangle = k(x_i, x_j)$, and spheres are constructed in the feature space $\mathbb{F}$ and they may represent highly complex shapes in the input space $\mathbb{R}^d$:

$$\{x : R^2 = k(x, x) - 2\sum_{i=1}^{n} \alpha_i k(x, x_i) + \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j)\} \ , \tag{9}$$

depending on one's choice of the kernel function k. Kernels that have proven to be effective for data description include the Gaussian kernel $k(x_1, x_2) = \exp(-\|x_1 - x_2\|^2/\sigma^2)$ and the polynomial kernel $k(x_1, x_2) = (1 + \langle x_1, x_2 \rangle)^p$.

3 Pattern Classification via Single Spheres

In the above section, we have described how to construct a minimum bounding sphere to provide a compact description of a set of data, which are assumed to belong to the same class. For each class, such a sphere can be constructed without considering training data from other classes. In this section, we explore the possibility of using single spheres for pattern separation.

Given a set of training data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$, instead of trying to find a sphere that provides a compact description of one class, for classification purposes, we want to find a sphere that encloses all examples from one class but excludes all examples from the other class, e.g., a sphere S with center c and radius R that encloses all positive examples and excludes all negative examples. In addition, we assume that sphere S separates the two classes with margin $2d$, i.e., it satisfies the following constraints:

$$R^2 - \langle x_i - c, x_i - c \rangle \geq d^2, \forall i \text{ such that } y_i = 1, \tag{10}$$

and

$$\langle x_i - c, x_i - c \rangle - R^2 \geq d^2, \forall i \text{ such that } y_i = -1, \tag{11}$$

where d is the shortest distance from the sphere to the closest positive and negative examples (see Fig. 1).

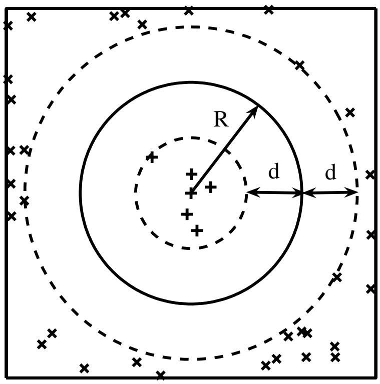

Fig. 1. Spherical classifier that maximizes the separation ratio

There may exist many spheres that satisfy the above constraints. Among many such spheres, it is natural that we seek to find a sphere that separates the training data with the maximum separation ratio, i.e.,

$$\max_{c,R,d} \frac{R+d}{R-d} \tag{12}$$

subject to

$$y_i(R^2 - \langle x_i - c, x_i - c\rangle) \geq d^2 \quad \forall i = 1, \ldots, n \ . \tag{13}$$

It is easy to show that maximization of the separation ratio $(R+d)/(R-d)$ is equivalent to minimization of R^2/d^2. The objective function R^2/d^2 is a nonlinear function of R^2 and d^2 and is hard to deal with directly. However, at any given point (R_0, d_0), R^2/d^2 can be approximated as:

$$\frac{R^2}{d^2} \approx \frac{R_0^2}{d_0^2} + \frac{1}{d_0^2}(R^2 - \frac{R_0^2}{d_0^2} d^2) \ . \tag{14}$$

Therefore, the problem of finding the sphere with maximum separation ratio can be reformulated as:

$$\min_{c,R,d} \ R^2 - Kd^2 \tag{15}$$

subject to

$$y_i(R^2 - \langle x_i - c, x_i - c\rangle) \geq d^2 \quad \forall i = 1, \ldots, n \ , \tag{16}$$

where $K = R_0^2/d_0^2 \geq 1$ is a constant that controls the ratio of the radius to the separation margin.

Introducing Lagrange multipliers $\alpha_i \geq 0$, one for each of the constraints in (16), we obtain the Lagrangian:

$$L = R^2 - Kd^2 - \sum_{i=1}^{n} \alpha_i [y_i(R^2 - \langle x_i - c, x_i - c\rangle) - d^2] \ . \tag{17}$$

The task is to minimize the Lagrangian L with respect to R, d, and c, and to maximize it with respect to α_i. Setting the partial derivatives to zero, we obtain

$$c = \sum_{i=1}^{n} \alpha_i y_i x_i \ , \tag{18}$$

which gives the center c of the sphere as a linear combination of training data x_i, and

$$\sum_{i=1}^{n} \alpha_i = K \tag{19}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 1 \ . \tag{20}$$

Substituting the new constraints into the Lagrangian (17), we obtain the following dual form of the quadratic programming problem:

$$\min_{\alpha_i, i=1,\dots,n} \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle - \frac{1}{2} \sum_{i=1}^{n} \alpha_i y_i \langle x_i, x_i \rangle \tag{21}$$

subject to

$$\alpha_i \geq 0, \quad \forall i = 1, \dots, n \tag{22}$$

$$\sum_{i=1}^{n} \alpha_i = K \tag{23}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 1 \ . \tag{24}$$

It should be emphasized that, unlike the quadratic programming problems in Sect. 2 or in standard SVMs, the primal constrained optimization problem defined by (15) and (16) is non-convex. In fact, it is easy to see that the set of constraints (16) for all i such that $y_i = -1$ is non-convex. However, fortunately, the Lagrangian (17) is convex at the solution of the dual problem. Therefore, strong duality still holds and the solution of the dual problem provides an optimal solution of the primal problem.

Solving the dual problem, one obtains the coefficients $\alpha_i, i = 1, \dots, n$. The center c of the optimal sphere can be obtained by Eq. (18). Similarly, the radius R can be determined from the KKT conditions by letting

$$R^2 = \frac{\min_{y_i=-1} \langle x_i - c, x_i - c \rangle + \max_{y_i=1} \langle x_i - c, x_i - c \rangle}{2} \ , \tag{25}$$

which leads to the following spherical decision function:

$$f(x) = \operatorname{sgn}\left(R^2 - \left(\langle x, x \rangle - 2 \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle + \sum_{i,j} \alpha_i \alpha_j \langle x_i, x_j \rangle \right) \right) \ . \tag{26}$$

In general, the solution to the above optimization problem may not exist because there is no such sphere in the input space that separates all the positive samples from the negative samples. Similarly to the SVDD case, we can apply the kernel trick here by replacing the inner products with suitable kernel functions. In effect, the maximum separation sphere is constructed in the feature space induced by the kernel. So far, we have only considered the case in which the data is separable by a sphere in the input space or in the feature space that is induced by the kernel. However, such a sphere may not exist, even in the kernel feature space. To allow for some classification errors, we introduce slack-variables $\xi_i \geq 0$ for $i = 1, \ldots, n$ to relax the constraints (13) with

$$y_i(R^2 - \langle x - c, x - c\rangle) \geq d^2 - \xi_i \ , \tag{27}$$

and consequently minimize the following objective function:

$$\min_{c,R,d,\xi_i,i=1,\ldots,n} R^2 - Kd^2 + C\sum_{i=1}^{n} \xi_i \ , \tag{28}$$

where the regularization constant C determines the trade-off between the empirical error and spherical separation margin term. Using the Lagrange multiplier method, we obtain the following dual problem in the kernel form:

$$\min_{\alpha_i,i=1,\ldots,n} \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i\alpha_j y_i y_j k(x_i, x_j) - \frac{1}{2}\sum_{i=1}^{n} \alpha_i y_i k(x_i, x_i) \tag{29}$$

subject to

$$0 \leq \alpha_i \leq C, \quad \forall i = 1, \ldots, n \tag{30}$$

$$\sum_{i=1}^{n} \alpha_i = K \tag{31}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 1 \ . \tag{32}$$

The above dual optimization problem can be solved using standard quadratic programming solvers, such as CPLEX, LOQO, MINOS and Matlab QP routines. Similarly to the standard SVMs, one can also use the sequential minimal optimization (SMO) method or other decomposition methods to speed up the training process by exploiting the sparsity of the solution and the KKT conditions [13,14,15].

It should be noted that separating data using spheres is a special case of separating data via ellipsoids, which results in a convex semi-definite program (SDP) that can be efficiently solved by interior point methods [16]. However, a drawback of the ellipsoid separation approach is that it cannot be easily extended by the kernel method, because the SDP problem cannot be expressed purely in inner products between input vectors. Therefore, both the decision boundaries it can generate and the problems it can solve are limited, unless special preprocessing is carried out prior to applying the ellipsoid separation method. On the other hand, using spheres combined with suitable kernels can produce more flexible decision boundaries than ellipsoids. Furthermore, SDP is limited in terms of the number of input dimensions it can effectively deal with.

4 Results and Discussion

We applied the method to both artificial and real-world data. The training algorithm was implemented based on the SMO method. Figure 2 displays a 2-D toy example and shows how different values of the parameter K lead to different solutions. The training examples of two classes are denoted as $+$'s and $\times$'s respectively in the figure. Clearly, there exist many spheres that can separate the training data in the 2-D input space. Therefore, for this dataset, no kernel trick was used, and the separating spheres were constructed directly in the input space using the standard definition of the Euclidean inner product. The three remaining plots show the results with three different values of the constant K. In each plot, three spheres (or their portions) are displayed. The darkest line represents the sphere with radius $R - d$. The lightest line represents the sphere with radius $R + d$. The line in between represents the separating sphere with radius R. The support vectors (the training examples with nonzero α values) are marked with small circles.

As we can see, increasing the value of K from 1 to 100, the shape of the decision surface changes from a sphere to a plane. When K is set to a small value, the algorithm finds a sphere that gives a compact description of the positive examples. For instance, when $K = 1$, the inner sphere (the sphere with radius $R - d$) coincides with the smallest sphere found by the SVDD method that encloses all the positive examples [12,17]. When K is set to a larger value, a larger sphere is found to contain the positive examples and the decision surface is more like a plane. Therefore, by adjusting the constant K that controls the ratio of the radius of the sphere to the separation margin, one can obtain a series of solutions from sphere-like decision boundaries to linear decision boundaries, including the solution of the SVDD method for data description and the solution of SVMs for classification.

Figure 3 shows the results of the spherical classifiers with a Gaussian kernel on another artificial dataset. The training data is generated randomly in a rectangular region. Training examples of the two classes, separated by $f_2 = \sin(\pi f_1)$, are denoted as $+$'s and $\times$'s respectively (see figure 3, upper-left plot). Clearly, there is no single sphere in the 2-D input space that can separate the two classes. We used a Gaussian kernel to map the data into a high dimensional feature space, in which the separating spheres were constructed. The remaining three plots show the results of the spherical classifier at different values of K. For better visualization, only training examples that correspond to the support vectors are shown in the three plots. The results demonstrate that a separating sphere was found in the feature space by adjusting the value of the constant K.

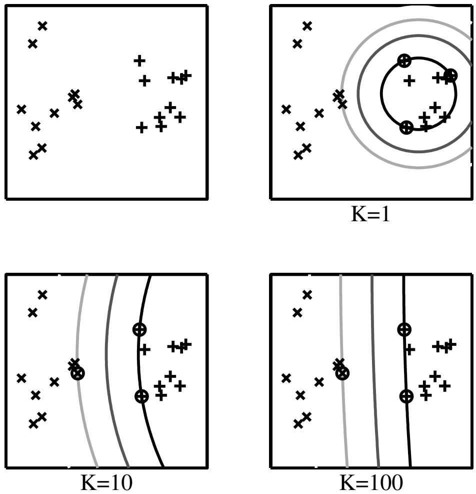

Fig. 2. Results of the spherical classifier on an artificial dataset at different values of K

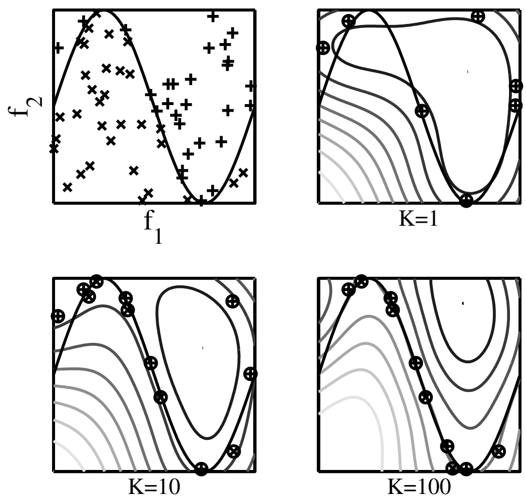

Fig. 3. Results of the spherical classifier (using a Gaussian kernel) on an artificial dataset. Top left: The training data and desired decision boundary; The rest: spheres of different radii mapped back onto the 2-D input space for three different values of K. The darker the line, the smaller the radius. The small circles around training examples indicate the support vectors.

We also tested the new algorithm and compared it to standard SVMs using several real-world datasets from the UCI machine learning repository [18]. For all the datasets, we used the 5-fold cross-validation method to estimate the generalization error of the classifiers. In the 5-fold cross-validation process, we ensured that each training set and each testing set were the same for both algorithms, and the same Gaussian kernel was used. The datasets used and the results obtained by the two algorithms are summarized in Table 1. The results of the spherical classifier and the SVM classifier both depend on the values of the kernel parameter σ and the regularization parameter C. In addition, the performance of the spherical classifier also depends on the value of K. In our tests, we set C to infinity for both algorithms, i.e., we only considered hard-margin spherical and hyperplane classifiers. On each dataset, the value of the kernel parameter σ was optimized to provide the best error rate of the SVM classifier, and the same value was used for the spherical classifier. As we can see, the spherical classifier achieves the same or slightly better results than SVMs on all 5 datasets.

Table 1. Comparison of Error Rates

Dataset	Sphere	SVM
Breast Cancer	4.26 (±1.73)	4.26 (±1.73)
Ionosphere	5.71 (±2.80)	6.00 (±2.86)
Liver	35.36 (±1.93)	36.23 (±5.39)
Pima	34.90 (±2.13)	35.03 (±2.20)
Sonar	10.73 (±1.91)	11.22 (±2.44)

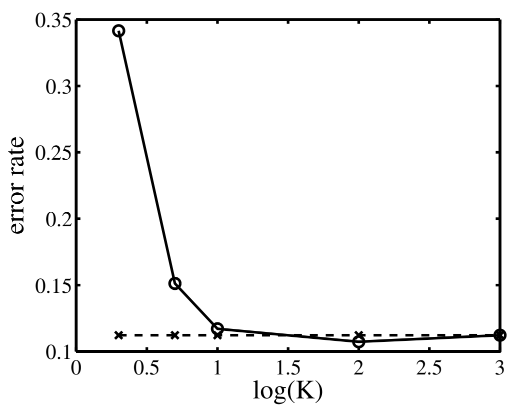

Fig. 4. Error rates of the spherical classifier on the sonar dataset for different values of K. The solid line represents the error rate of the spherical classifier. The dashed line is the error rate of the SVM classifier.

In Figure 4, we show a detailed comparison of the spherical classifier and the SVM classifier on the Sonar dataset. The solid line displays the error rates of the spherical classifier at different values of K. The dashed line gives the corresponding error rates of the support vector machine. Once again, the same kernel parameter σ was used for both algorithms, and the regularization parameter C was set to infinity. As we can see, the error rates of the spherical classifier decrease as the value of K increases. If K is set to be large enough, the result of the spherical classifier reaches that of the support vector machine, which is consistent with what we have observed in our toy examples.

From Table 1 and Fig. 4, we see that the spherical classifier yields comparable results as the support vector machine, demonstrating that it is suitable for real-world classification problems.

5 Conclusion

In this paper we explored the possibility of using single spheres for pattern classification. Inspired by the support vector machines and the support vector data description method, we presented an algorithm that constructs single spheres in the kernel feature space that separate data from different classes with the maximum separation ratio. By incorporating the class information of the training data, our approach provides a natural extension to the SVDD method of Tax and Duin, which computes minimal bounding spheres for data description (also called One-class classification).

By adopting the kernel trick, the new algorithm effectively constructs spherical boundaries in the feature space induced by the kernel. As a consequence, the resulting classifier can separate patterns that would otherwise be inseparable when using a single sphere in the input space. Furthermore, by adjusting the ratio of the radius of the separating sphere to the separation margin, a series of solutions ranging from spherical to linear decision boundaries can be obtained. Specifically, when the ratio is set to be small, a sphere is constructed that gives a compact description of the positive examples, coinciding with the result of the SVDD method; when the ratio is set to be large, the solution effectively coincides with the maximum margin hyperplane solution. Therefore, our method effectively encompasses both the support vector machines for classification and the SVDD method for data description. This feature of the proposed algorithm may also be useful for dealing with the class-imbalance problem. We tested the new algorithm and compared it to the support vector machines using both artificial and real-world datasets. The experimental results show that the new algorithm offers comparable performance on all the datasets tested. Therefore, our algorithm provides an alternative to the maximum margin hyperplane classifier.

References

1. Cooper, P.W.: The hypersphere in pattern recognition. Information and Control **5** (1962) 324–346
2. Cooper, P.W.: Note on adaptive hypersphere decision boundary. IEEE Transactions on Electronic Computers (1966) 948–949

3. Batchelor, B.G., Wilkins, B.R.: Adaptive discriminant functions. Pattern Recognition, IEEE Conf. Publ. **42** (1968) 168–178
4. Batchelor, B.G.: Practical Approach to Pattern Classification. Plenum, New York (1974)
5. Reilly, D.L., Cooper, L.N., Elbaum, C.: A neural model for category learning. Biological Cybernetics **45** (1982) 35–41
6. Scofield, C.L., Reilly, D.L., Elbaum, C., Cooper, L.N.: Pattern class degeneracy in an unrestricted storage density memory. In Anderson, D.Z., ed.: Neural Information Processing Systems. American Institute of Physics, Denver, CO (1987) 674–682
7. Marchand, M., Shawe-Taylor, J.: The set covering machine. Journal of Machine Learning Research **3** (2002) 723–746
8. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In Haussler, D., ed.: Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory. (1992) 144–152
9. Cortes, C., Vapnik, V.N.: Support vector networks. Machine Learning **20** (1995) 273–297
10. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York, NY (1998)
11. Schölkopf, B., Burges, C., Vapnik, V.: Extracting support data for a given task. In: Proceedings of First International Conference on Knowledge Discovery and Data Mining. (1995) 252–257
12. Tax, D.M.J., Duin, R.P.W.: Data domain description by support vectors. In Verleysen, M., ed.: Proceedings ESANN, Brussels, D. Facto Press (1999) 251–256
13. Platt, J.: Fast training of support vector machines using sequential minimal optimization. In Schölkopf, B., Burges, C.J.C., Smola, A.J., eds.: Advances in Kernel Methods - Support Vector Learning, Cambridge, MA, MIT Press (1999) 185–208
14. Vapnik, V.N.: Estimation of Dependence Based on Empirical Data. Springer-Verlag, Berlin (1982)
15. Osuna, E., Freund, R., Girosi, R.: Support vector machines: training and applications, A.I. Memo AIM - 1602. MIT A.I. Lab (1996)
16. Glineur, F.: Pattern separation via ellipsoids and conic programming, Mémoire de D.E.A., Faculté Polytechnique de Mons, Mons, Belgium (Sept. 1998)
17. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J.: Estimating the support of a high-dimensional distribution. Neural Computation **13** (2001) 1443–1471
18. Blake, C., Merz, C.: UCI repository of machine learning databases, http://www.ics.uci.edu/∼mlearn/MLRepository.html (1998)