



Hyper-Sphere Support Vector Classifier with Hybrid Decision Strategy

Shuang Liu¹ and Peng Chen²(✉)

¹ Dalian Minzu University, Dalian 116600, China
dlnuliushuang@qq.com

² Dalian Neusoft University of Information, Dalian 116023, China
44531347@qq.com

Abstract. If all bounding hyper-spheres for training data of every class are independent, classification for any test sample is easy to compute with high classification accuracy. But real application data are very complicated and relationships between classification bounding spheres are very complicated too. Based on detailed analysis of relationships between bounding hyper-spheres, a hybrid decision strategy is put forward to solve classification problem of the intersections for multi-class classification based on hyper-sphere support vector machines. First, characteristics of data distribution in the intersections are analyzed and then decision class is decided by different strategies. If training samples of two classes in the intersection can be classified by intersection hyper-plane for two hyper-spheres, then new test samples can be decided by this plane. If training samples of two classes in the intersection can be approximately linearly classified, new test samples can be classified by standard optimal binary-SVM hyper-plane. If training samples of two classes in the intersection cannot be linearly classified, new test samples can be decided by introducing kernel function to get optimal classification hyper-plane. If training examples belong to only one class, then new test samples can be classified by exclusion method. Experimental results show performance of our algorithm is more optimal than hyper-sphere support vector machines with only one decision strategy with relatively low computation cost.

Keywords: Hyper-sphere support vector classifier · Intersection region · Hybrid decision strategy · Classification hyper-plane · Multi-class classification

1 Introduction

Support Vector Machine (SVM) is originally put forward to solve binary classification problem with the idea of maximum-margin hyper-plane separating training samples. The classification hyper-plane is restricted with support vectors, which are used for the decision of a new test sample. Many researchers have successfully applied SVM in many fields [1, 2]. To solve multi-class classification

problem in real applications, binary SVM needs to be extended or combined together to complete the complex classification task. There are many methods extending binary SVM to multi-class classification [3–5], such as one-against-one, one-against-all, hierarchy SVM classifiers, or DAG SVM classifiers. But because at least $k(k-1)/2$ quadratic programming (QP) optimization problems needs to be combined together to solve k class classification problem, computation cost for these methods is relatively high.

For multi-class classification problem, sphere-structured SVM is one special solution [6]. Based on one-class SVM, hyper-sphere SVM classifier tries to construct a minimum bounding hyper-sphere restricting all training samples of one class within it as much as possible. The bounding hyper-sphere for each class is restricted with its center and its radius. Similar to binary SVM, this method maximizes the gap between different hyper-spheres by the smallest radius. New test samples are classified depending on the bounding hyper-spheres they falls into. Since this method needs no combination of further computation and solves multi-class classification problem with direct computation of all hyper-spheres together, its computation complexity is less than all the above mentioned combination methods. Sphere-structured SVM has been studied a lot since its presentation. To get good performance for hyper-sphere SVM, Liu et al. [7, 8] proposed one fuzzy hyper-sphere SVM and one multiple sub-hyper-spheres SVM for multi-class classification. Most of these research focused on one decision rule and few researchers adopted two or more decision rules.

As mentioned above, each training data is bounded within a bounding hyper-sphere. If all bounding hyper-spheres for training data of every class are independent and new test sample falls inside only one hyper-sphere, its classification decision is easy to compute. But real application data are very complicated and relationships between different classification bounding hyper-spheres are very complicated too. When hyper-spheres for each class are intersected or not independent, it is difficult for one simple decision function to get the right classification result. Classification accuracy of test samples falling inside the intersection will influence the final classification performance. Based on the analysis of data distributions for such samples in the intersections, a hybrid decision strategy is put forward in this paper. Section 2 introduces statistical analysis of the intersection data distribution, mathematical description of our hyper-sphere SVM and implementation details of the proposed method. Section 3 discusses the experimental results and Sect. 4 gives the conclusions.

2 Our Method

2.1 Mathematical Description of Hyper-Sphere Support Vector Classifier

Similar with mathematic description of the original binary SVM, mathematic principles of hyper-sphere SVM is as follows. Supposing there is a set of n -dimensional training samples of m classes, the task is to compute the minimum bounding hyper-sphere for each class. Here, the minimum bounding hyper-sphere

refers to the smallest hyper-sphere that encloses all the training samples of one class. Referenced by binary SVM, slack variables ξ_{ki} are introduced by permitting isolated points and a non-linear mapping function ϕ is introduced by transforming the training samples into a high dimensional feature space to solve nonlinear separation problems. So seeking the minimum bounding hyper-sphere for each class is to find the minimum bounding hyper-sphere enclosing all the training examples of that class. This process can be computed by solving the following constrained quadratic optimization problem in Eq. (1).

$$\begin{aligned} \min_{c_k, R_k} R_k^2 + C_k \sum_{i=1}^m \xi_{ki} \\ \text{s.t.} \\ \|\phi(\mathbf{x}_i) - c_k\|^2 \leq R_k^2 + \xi_k \\ \xi_i \geq 0, i = 1, \dots, l_k \end{aligned} \quad (1)$$

For class k , its minimum bounding hyper-sphere S_k is characterized by its center c_k and radius R_k . And C_k is the penalty factor and $\xi_{ki} \geq 0$ are slack variables.

By introducing Lagrange multipliers, Lagrange polynomial can be written as Eq. (2).

$$L(R_k, c_k, \xi_i, \alpha_i, \beta_i) = R_k^2 + C_k \sum_{i=1}^{l_k} \xi_{ki} - \sum_{i=1}^{l_k} \alpha_i (R_k^2 + \xi_{ki} - \|\phi(\mathbf{x}_i) - c_k\|^2) - \sum_{i=1}^{l_k} \beta_i \xi_{ki} \quad (2)$$

By taking the partial directives of L with respect to R_k, c_k and ξ_i and substituting them back to Eq. (2), the original optimization problem becomes its dual optimization problem in the following format as Eq. (3).

$$\begin{aligned} \min_{\alpha_i} \sum_{i,j=1}^{l_k} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{l_k} \alpha_i K(\mathbf{x}_i, \mathbf{x}_i) \\ \text{s.t.} \\ \sum_{i=1}^{l_k} \alpha_i = 1 \\ 0 \leq \alpha_i \leq C_k, i = 1, \dots, l_k \end{aligned} \quad (3)$$

In Eq. (3), the kernel trick is adopted to compute inner products in the feature space, that is, $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$. Support vectors are the vectors \mathbf{x}_i with $\alpha_i > 0$. The Lagrange multipliers get the solutions after solving Eq. (3). So the center can be computed by Eq. (4) and the resulting decision function can be computed as Eq. (5). Then the radius R_k can be computed by equating $f_k(\mathbf{x})$ to zero for any support vector. For class k , its minimum bounding hyper-sphere S_k is obtained by the solution of its center and radius.

$$c_k^2 = \sum_{i,j=1}^{l_k} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (4)$$

$$f_k(\mathbf{x}) = \text{sgn}(R_k^2 - \sum_{i,j=1}^{l_k} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)) + 2 \sum_{i=1}^{l_k} K(\mathbf{x}_i, \mathbf{x}) - K(\mathbf{x}, \mathbf{x}) \quad (5)$$

Based on Eq. (5), the new point \mathbf{x} falls inside of the hyper-sphere if $f_k(\mathbf{x}) > 0$. \mathbf{x} falls outside of the hyper-sphere if $f_k(\mathbf{x}) < 0$ and \mathbf{x} lies on the hyper-sphere if $f_k(\mathbf{x}) = 0$.

2.2 Analysis of Data in the Intersection of Hyper-Spheres

Ideally, all hyper-spheres are independent and each test sample is correctly classified by one hyper-sphere decision function. But it can happen that two or more hyper-spheres intersect, that is, one sample falls inside several hyper-spheres. Or a new test sample falls outside of all hyper-spheres. How to correctly classify these data points influence the accuracy performance of the resulting classifier. In Fig. 1, the minimum bounding sphere for class 1 and class 2 are S_1 and S_2 . S'_1 of class 1 belongs to sphere S_2 and S'_2 of class 1 belongs to sphere S_1 . \mathbf{x}_1 belongs to S'_1 and \mathbf{x}_2 belongs to S'_2 and they are both support vectors. For example, \mathbf{x}_2 is a support vector of class 2, so \mathbf{x}_2 belongs to class 2. But based on the decision function, $f_2(\mathbf{x}_2) = 0$ and $f_1(\mathbf{x}_2) > 0$, so \mathbf{x}_2 belongs to class 1. Obviously, it is the wrong classification result. So when two or more hyper-spheres intersect, only Eq. (5) is used as decision rule may lead to wrong decision results.

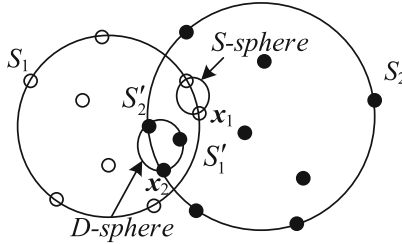


Fig. 1. Illustration of two hyper-spheres intersecting.

To solve this problem, one sub-hyper-sphere support vector machine is put forward to classify samples in the intersections in [9]. In the research, same error data hyper-sphere (data points belong to the same class of the mother hyper-sphere) and different error data hyper-sphere (data points belong to different class of the mother hyper-sphere) are introduced. The decision process is completed by them. In Fig. 1, S-sphere and D-sphere are same error data sub-hyper-sphere and different error data hyper-sphere for class 1. If a new test sample lies in the intersection, S-sphere and D-sphere are used as the classification rule to get the right class.

But by introducing multiple QP optimization problems again, its computation complexity increases. To reduce computation complexity of QP optimization problem, new decision rules are put forward in this paper. Given the training data set, there will be three cases for position of the new test sample, that is, inside one hyper-sphere or in the intersections or outside of all spheres after computing hyper-spheres for all classes. It is easy to get decision class for the inclusion case. If a new test sample \mathbf{x}_2 falls outside of all spheres, Eq. (6) is adopted as its class j decision ($j = 1, \dots, m$).

$$\min\left(\frac{d_{x_j}^2}{R_j^2} - 1\right), \text{ where } d_{x_j}^2 = \|\phi(\mathbf{x}) - c_j\|^2 = K(\mathbf{x}, \mathbf{x}) - 2 \sum_{i=1}^{l_k} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + \sum_{i,j=1}^{l_k} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (6)$$

If a new test sample belongs to the intersections, there are three cases for different data distribution as following.

(1) For the first case, the intersection hyper-plane can separate samples of two classes directly. Intersection hyper-plane is easy to get from subtraction of two spheres equations.

(2) For the second case, intersection hyper-plane cannot separate samples of two classes directly, so binary optimal plane is used as separation plane for linear and nonlinear cases as shown in Fig. 2.

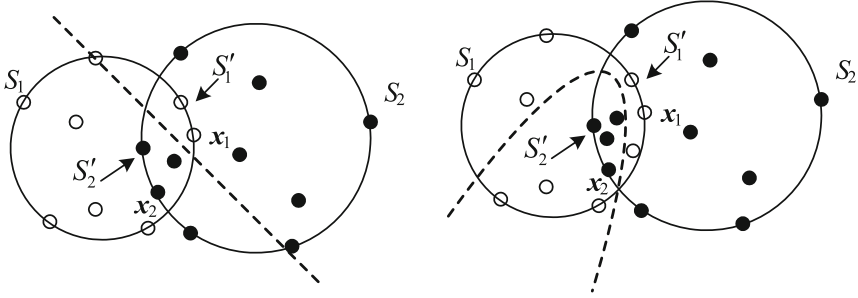


Fig. 2. Case of binary classification plane as separation plane for linear (left) and nonlinear (right) data.

(3) For the third case, there is only one class data in the intersection. For this case, exclusion method is adopted as the decision rule.

As can be seen from the reference [6,9], training time complexity for the sphere-structured SVM is $O(n^2)$. Testing time for the sphere-structured SVM is decided by its decision process. For our new hybrid hyper-sphere SVM, its training time complexity is $O(n^2)$. In testing phase, time complexity may be $O(1)$ for case of only one class data in the intersection, $O(n)$ for case of binary classification plane as separation plane and $O(n^2)$ for case of nonlinear classification plane as separation plane. The average time complexity is lower than that of sub-hyper-sphere SVM with $O(n^2)$.

2.3 Implementation Details of the Proposed Method

The proposed hyper-sphere support vector classifier with hybrid decision strategy follows three steps to complete classification process, which are listed as Algorithm 1. And testing process is described as Algorithm 2. Suppose there are m classes needs to be classified.

3 Experimental Results and Analysis

In this section some experimental results are given to verify the efficiency of the new improved classification rule and compare it with hyper-sphere SVMs and sub-hyper-sphere SVMs. The first experimental datasets IRIS and glass come from the UCI [10] Repository of machine learning databases. The second data comes from one real application of action recognition in videos.

For three classes of IRIS data set, all samples are randomly grouped as three parts and one part with number 15 is used as test, the other two 135 as training set. For glass data set, a subset with three classes is selected and the same data pre-processing is adopted as paper [6]. For hyper-sphere SVMs and sub-hyper-sphere SVMs [9], $C = 100, \sigma = 1.25$ is used. For our new approach, RBF kernel with parameter optimization search are used to get the best classification accuracy. Experimental results are shown in Table 1. All experiments were done for 10 times and data in Table 1 is the average value of each indicator. Experimental results show our new hybrid classification rule is effective and easy to compute for the simple multi-class classification problems.

The second experimental data set is one action recognition system from videos collecting from the Internet. Extracted features consist of foregrounds extraction, morphology operations and shape feature, KLT tracking points and so on. These features are input into the proposed hybrid hyper-sphere SVM classifiers. The dimension of the input image is 20 dimensions, the number of categories is 6, and the size of data set is 9000. Among 9000 samples, data1 consists of 3000 randomly selected samples for each class and 600 testing samples, and data2 with 3000 and 600, data3 with 7800 and 1200 as data1. The experiment was repeated 10 times and the average value was selected as the final experimental result. The accuracy of our method is 86.15, 90.11 and 92.15 for three datasets respectively. For the other two methods, the highest accuracy is fuzzy hyper-sphere SVM, the values are 85.04, 89.19 and 90.67 for three datasets and 83.98, 88.30, 89.33 for sub-sphere SVM. Performance of our new method is better than the other two methods.

Algorithm 1: Training Process for our HSSVM classifier with hybrid decision strategy

Step1 reads training data for each class.

for $k=1$ to m **do**

 QP optimization problems are solved based on SMO algorithm, Lagrange multipliers are obtained by QP solving, and sphere model is obtained based on Lagrange multipliers;

end

Step2 computes position relations between any two spheres.

for $i=1$ to m **do**

for $j=i+1$ to m **do**

$D_{i,j} = \|c_i - c_j\|^2$ (Euclidean distance between center of the i -th and j -th sphere);

if $|R_i - R_j| \leq D_{i,j} \leq |R_i + R_j|$ **then**
 | hyper-sphere S_i, S_j intersects and saves in SR_i ;

end

if $D_{i,j} > |R_i - R_j|$ **then**
 | hyper-sphere S_i, S_j are independent and saves SR_d ;

end

end

end

Step3 sets decision rules for each pair of spheres in SR_i . For class j , its corresponding hyper-sphere is S_j . Here, S'_j is the intersection of S_i and S_j . C_j is data set of different error samples in S'_j and A_j is data set of same error samples in S'_j .

if $|C_j| = 0$ **and** $|A_j| = 0$ **then**

 all data in the intersection belongs to one class and saves it as the exclusion file; If new test sample falls in the intersection, its class decision is the class with number of training samples larger than zero;

end

if $|C_j| \neq 0$ **and** $|A_j| \neq 0$ **then**

if $|C_j| > 1$ **and** $|A_j| > 1$ **then**

 intersecting plane is computed for two spheres S_i and S_j . If samples of two classes can be separated directly by the intersecting plane, it is saved as decision classification plane. If data of two classes falls in two sides of the intersecting plane, this intersecting plane is saved as decision plane;

end

if *samples of two classes can not be separated directly by the intersecting plane* **then**

 binary optimal SVM plane by introducing slack variables or map function is computed and saved as the decision plane;

end

if $|C_j| == 1$ **and** $|A_j| == 1$ **then**

 two spheres are tangent and two classes are saved as classification result;

end

end

Algorithm 2: Testing Process for our HSSVM classifier with hybrid decision strategy

```

//computes classification results for test samples
for  $i=1$  to  $M$  do
  for  $j=1$  to  $m$  do
    distance  $d_{ij}^2$  between sample  $i$  and  $S_j$  is computed;
    if  $d_{ij}^2 - R_j^2 \leq 0$  then
      | sample  $i$  falls inside of  $S_j$ ;
    else
      | sample  $i$  falls outside of the sphere;
    end
  end
  number  $N$  of hyper-spheres containing sample  $i$  is counted;
  if  $N == 1$  then
    | sample  $i$  falls inside only one sphere, it belongs to this class;
  end
  if  $N == 0$  then
    |  $i$  falls outside of all spheres, use Eq.(6) as decision rule;
  end
  if  $N > 1$  then
    Suppose sample  $i$  falls inside  $S_j, j = 1, \dots, r$ ;
    for  $j=1$  to  $r$  do
      for  $e=j+1$  to  $r$  do
        if  $S_j, S_e \in SR_i$  then
          if binary optimal plane exists then
            | this binary optimal plane is used as decision plane;
          end
          if binary optimal plane exists then
            | exclusion method is adopted as decision rule;
          end
          if  $i \in \text{spherical surface of } S_j, S_e$  then
            | sample  $i$  belongs to two classes;
          end
        end
      end
    end
  end
end
end
end

```

Table 1. Comparison result for IRIS classification.

Iris/glass	Training time	No. of Support Vector	Accuracy
Traditional hyper-sphere SVM	4.56 s/5.89 s	45/54	96.68%/62.15%
Sub-hyper-sphere SVM	5.78 s/7.35 s	56/68	97.78%/65.96%
New hybrid hyper-sphere SVM	3.556 s/4.785 s	37/42	99.86%/68.75%

4 Conclusions

To improve classification performance of traditional hyper-sphere SVM, one hybrid decision strategy for hyper-sphere support vector classifier is put forward in this paper. To get high classification performance for test samples and decreases computation complexity of QP optimization problems, four decision rules are discussed and detailed algorithm is given. Results on benchmark data and real application data show our hybrid decision rule leads to better generalization accuracy than the existing methods, decreasing computation complexity and saving training time.

Acknowledgement. This research was funded by the Natural Science Foundation of Liaoning Province, China (grant no. 2019-ZD-0175).

References

1. Mohanty, S.: Speaker identification using SVM during oriya speech recognition. *Int. J. Image, Graph. Signal Process.* **10**, 28–36 (2015)
2. Sun, A., Lim, E.P., Liu, Y.: On strategies for imbalanced text classification using SVM: a comparative study. *Decis. Support Syst.* **48**(1), 191–201 (2009)
3. Hsu, C.W., Lin, C.J.: A comparison of methods for multi-class support vector machines. *IEEE Trans. Neural Netw.* **13**(2), 415–425 (2002)
4. Vural, V., Dy, J.G.: A hierarchical method for multi-class support vector machines. In: *ACM International Conference on Machine Learning*, pp. 831–838 (2004)
5. Chmielnicki, W., Stapor, K.: Combining one-versus-one and one-versus-all strategies to improve multiclass SVM classifier. In: *Proceedings of the 9th International Conference on Computer Recognition Systems*, pp. 37–45 (2016)
6. Zhu, M., Wang, Y., Chen, S., Liu, X.: Sphere-structured support vector machines for multi-class pattern recognition. In: Wang, G., Liu, Q., Yao, Y., Skowron, A. (eds.) *RSFDGrC 2003. LNCS (LNAI)*, vol. 2639, pp. 589–593. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39205-X_95
7. Liu, S., Chen, P., Li, K.Q.: Multiple sub-hyper-spheres support vector machine for multi-class classification. *Int. J. Wavelets Multiresolut. Inf. Process.* **12**(3), 1450035 (2014)
8. Liu, S., Chen, P., Yun, J.: Fuzzy hyper-sphere support vector machine for pattern recognition. *ICIC Express Lett.* **9**(1), 87–92 (2015)
9. Wu, Q., Jia, C.Y., Zhang, A.F.: An improved algorithm based on sphere structure SVMs and simulation. *J. Syst. Simul. (Chinese)* **20**(2), 345–348 (2008)
10. UCI repository of machine learning databases. <http://www.ics.uci.edu/mllearn/>. Accessed 1 May 2020