A Classifier Based on Minimum Circum Circle

Xi Huang, Ying Tan, and Xingui He

Key Laboratory of Machine Perception and Intelligence (Peking University),
Ministry of Education, Department of Machine Intelligence,
School of Electronics Engineering and Computer Science,
Peking University, Beijing 100871, P.R. China
ytan@pku.edu.cn

Abstract. A new linear classifier based on minimum circum circle (CMCC) is proposed in this paper. It first calculates the minimum circum circle of samples for each class. Then the distributions of the samples can be described by these circles. The linear separating hyperplane will intersect the connecting line of the centers of circles. Consequently, the perpendicular to the connecting line of each two centers is defined as the classifier of these two classes. Moreover, some improved classifiers are proposed when the separating hyperplane is not perpendicular to the connecting line or when there are outliers in the samples. The combined classifier based on subclasses is also discussed.

In the experiments, the CMCC and its improved algorithms are compared with some other classifiers such as support vector machine, linear discriminant analysis, etc. The experimental results show that the CMCC gives a relatively good performance on both classification accuracy and time cost.

Keywords: classifier, minimum circum circle, machine learning.

1 Introduction

In the field of machine learning, the classification is a basic and important problem. Many machine learning algorithms are based on the classification. The goal of classification is to capture the characteristics of an object to discriminate its class. A linear classifier is a fast and valid approach to do this.

A linear classifier usually makes a classification decision based on the value of a linear combination of the characteristics. These characteristics are also called a feature vector because they are usually presented as a vector. So a linear classifier is in fact a linear function to map a feature vector to the class of the object. For a two-class classification problem, the operation of a linear classifier can be seen as splitting a high-dimensional input space with a hyperplane. The points on the two sides of the hyperplane are classified into two classes.

The linear function can also be seen as an inner product of weights vector and feature vector. Due to the different weight values, the classifiers will be different. So how to find a proper classifier to discriminate the object accurately is the key problem of classification. To solve the problem, the different learning algorithms are proposed. For a supervised learning, the linear function is learned from a set of labeled training samples. An unsupervised learning algorithm only approximates the function by the input

Y. Tan, Y. Shi, and Z. Ji (Eds.): ICSI 2012, Part II, LNCS 7332, pp. 74–83, 2012.

features. Semi-supervised learning combines both labeled and unlabeled samples to generate an appropriate function or classifier [1].

Due to its high speed, a linear classifier is often used in situations where the speed of classification is an issue. Besides, the linear classifier usually works well when the number of dimensions is large or when the dataset is sparse. As a result, the linear classifier is usually adopted in document classification [2], image recognition [3], virus detection [4], etc.

2 Related Works

Generally speaking, the linear classifier can be classified into two classes based on its algorithm to approximate the linear classifier.

The first class is based on the conditional density function. For this type of classifier, the algorithm will assume the probability distributions of different classes, and then it will discriminate the class of object based on the conditional density [5].

For example, linear discriminant analysis (LDA) assumes Gaussian conditional density models. Under the assumption, the optimal parameter of the classifier will be determined based on the training samples. Then the class of the object can be given by using the trained classifier and its feature vector [6].

Naive Bayes classifier (NBC) is similar to the LDA. It assumes independent binomial conditional density model for each feature. Depending on the probability model, the NBC can be trained by a supervised learning algorithm. [7].

The second class of classifier usually includes discriminative model, which attempts to maximize some object of classification on a training set.

Logistic regression (LR) is such a classifier. It assumes the observed samples are generated by a binomial model, and attempt to fit the data to a logistic function. It is a generalized linear model used for binomial regression [8].

Perceptron also belongs to this class. The perceptron is considered as the simplest feed-forward neural network. It attempts to reduce the classification errors generated in the training samples to revise the weight values of the classifier [9].

Support vector machine (SVM) is a widely used classifier. It constructs a separating hyperplane in a high-dimensional space. The optimal classifier will maximum the margin between the decision hyperplane and the training samples of two classes to lower the generalization error of the classifier. [10].

In this paper, the CMCC is proposed. It adopts the minimum circum circle to describe the samples of different classes. The points on the circle can be seen as the "support vector". Then the classifier is the combination of these support vectors. The supervised learning algorithm maximum the accuracy of the classification to generated the optimal classifier. Then the object can be discriminated.

The paper will be organized as follows. In Section 3, the classification problem will be defined. In Section 4, the algorithm of the CMCC will be presented. In Section 5, based on the basic algorithm, some improvements are proposed. In Section 6, the experimental results will prove the effectiveness of the CMCC. At last, Section 7 is the conclusion of this paper.

3 Problem Definition

For a two-class classification problem, we are given some training data $\mathcal{D}$ including n points as follows.

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathbb{R}^p, y_i \in \{-1, 1\}\}_{i=1}^n \tag{1}$$

where the y_i is either 1 or -1, indicating the class of the point $\mathbf{x}_i$. Each $\mathbf{x}_i$ is a p-dimensional real vector. We hope to find a hyperplane that separates the points having $y_i = 1$ and the ones having $y_i = -1$.

For a linear classifier, the hyperplane or the classifier can be expressed as

$$y_i = f(\mathbf{w} \cdot \mathbf{x}_i) = f(\sum_{j=1}^{p} w_j x_{ij}) \tag{2}$$

where $\mathbf{w}$ is a real vector of weights and $f(\cdot)$ is a function that converts the inner product of the two vectors into the desired output class. Generally, the weight vector $\mathbf{w}$ is obtained by maximizing the accuracy of the classification on the set of labeled training samples. This process is the learning algorithm of the classifier. In the next section, we will propose the algorithm based on the minimum circum circle to solve the problem.

4 Algorithm

To solve the above classification problem, we adopt the minimum circum circle to describe the samples of each class. The minimum circum circles can be seen as the boundary of the samples. And all the points of the class will be included in the circle. If we can find a separating hyperplane to divide the circle from the other, that will be the classifier. The points on the circular boundary is similar to the "support vector" to represent the samples. The classifier can be expressed by the combination of these "support vectors".

For a two-class classification problem in 2 dimensions, the learning algorithm of the CMCC is as follows.

- **Step 1:** Calculate the minimum circum circle of samples for each class and determine the centers of each minimum circum circle.
- **Step 2:** Calculate the connecting line passing through the two centers as follows. If the centers of two circles are (x_1, y_1), (x_2, y_2), respectively. Then the points on the connecting line can be expressed as

$$(\frac{x_1 + \lambda x_2}{1 + \lambda}, \frac{y_1 + \lambda y_2}{1 + \lambda}) \tag{3}$$

  where $\lambda > 0$ is a variable parameter. That means the intersecting points of perpendicular and the connecting line will be variable.
- **Step 3:** Calculate the perpendicular of the connecting line. Based on the above assumptions, the perpendicular will be expressed as

$$(x_1 - x_2)(x - \frac{x_1 + \lambda x_2}{1 + \lambda}) + (y_1 - y_2)(y - \frac{y_1 + \lambda y_2}{1 + \lambda}) = 0 \tag{4}$$

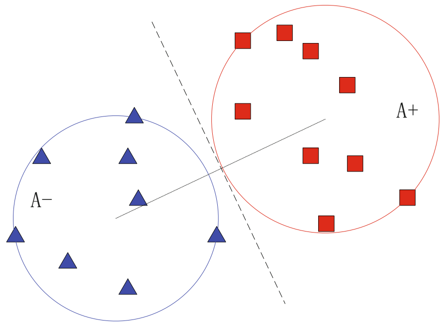

Fig. 1. A two-class classifier based on minimum circum circle

- **Step 4:** The above perpendicular which maximizing the accuracy of classification will be the classifier of two class.

As shown in Fig. 1, all the samples of two classes are included in its minimum circum circles. The dashed line is the classifier of the samples, which is perpendicular to the connecting line of two centers and classify the samples accurately.

The difficulty of the above algorithm is to calculate the minimum circum circle, so we give the algorithm as follows.

- **Step 1**: Select 3 points A, B, C from training samples randomly. We usually select the most distant two points as A and B, which can speed up the algorithm.
- **Step 2**: Calculate the center of minimum circum circle including the points A, B, C based on the analytic geometry method.
- **Step 3**: Search for a new point D which is most distant from the center of the minimum circum circle. If point D is in the circle or on the circle, end the algorithm; else, go to step 4.
- **Step 4**: Select 3 points from A, B, C and D, and generate a minimum circum circle including these 4 points. Redefine these 3 points as new points A, B and C, go to step 2. If there are only 2 of these 4 points on the circle, choose these 2 points as A and B and select one point randomly as C from the other 2 points.

To obtain the appropriate classifier, we tune λ to make the accuracy reach the highest. If the accuracy is the same on different λ, we adopt the λ which make the classifier farthest from the center of the minimum circum circle, which will lower the generalization error of the classifier.

In higher-dimensional case, the expression or form of classifier is similar to 2-dimensional problem. However, when the number of dimension is over 3, the minimum circum circles are difficult to calculate. We adopt the minimum circum rectangle to replace the circle. The center is substituted by the center of rectangle correspondingly. If the samples belong to some class are $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, the center of the minimum circum rectangle of this class will be

$$\mathbf{x}_o = \frac{1}{2}(\mathbf{x}_{\max} + \mathbf{x}_{\min}) = \frac{1}{2}(\max(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) + \min(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)) \quad (5)$$

where $\max(\mathbf{x})$ and $\min(\mathbf{x})$ indicate the vector containing the maximum value and minimum value of **x** in each dimension, respectively.

5 Improved Algorithm

5.1 Improved Algorithm 1

In the experiments, the perpendicular classifier can not always give a high classification accuracy. As a result, the constraint of the problem is extended. The angle between the classifier and the connecting line of two centers is set variable. We give a linear regression equation for the samples of each classes. The angle range of the classifier will be between this two regression line. The classifier can be expressed as follows.

For a two-class problem, if the linear regression equations are

$$a_1 x + b_1 y + c_1 = 0 \quad (6)$$

$$a_2 x + b_2 y + c_2 = 0 \quad (7)$$

the improved classifier will be

$$\frac{(a_1 + \eta a_2)}{1 + \eta}(x - \frac{x_1 + \lambda x_2}{1 + \lambda}) + \frac{(b_1 + \eta b_2)}{1 + \eta}(y - \frac{y_1 + \lambda y_2}{1 + \lambda}) = 0 \quad (8)$$

where $\eta \notin \{0, -1\}$ and $\lambda > 0$ are both variable parameters. They will control the angle and intersection of the classifier and the connecting line of two centers, respectively.

The improved classifier is presented as Fig. 2. In the figure, the samples of two classes can not be classified correctly based on the dashed line which is perpendicular to the connecting line of the centers. However, the improved classifier, which is shown as the solid line and not perpendicular to the connecting line of the centers, can discriminate the samples correctly.

5.2 Improved Algorithm 2

For most datesets, we could find some outliers in the training samples. The outlier is usually distant from the gravity center of the samples and will affect the performance of the classifier. So we use the distance between center of gravity and the minimum circum circle center as a parameter to improve the accuracy. If the distance between them exceed some limit, we drop the point farthest form the center of gravity and recalculate the minimum circum circle as follows.

We first calculate the center of minimum circum circle $\mathbf{s}_0$ based on the original algorithm of CMCC. Then the center of gravity $\bar{\mathbf{s}}$ should be calculated, where

$$\bar{\mathbf{s}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \quad (9)$$

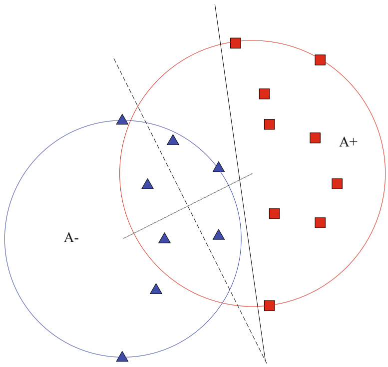

Fig. 2. An improved classifier based on minimum circum circle

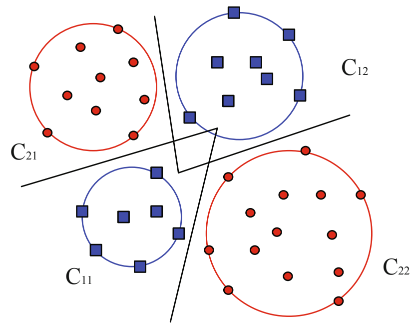

Fig. 3. A combined CMCC based on subclasses classifiers

We define the distance between these two centers as $d(\bar{\mathbf{s}}, \mathbf{s}_0)$. If

$$\frac{d(\bar{\mathbf{s}}, \mathbf{s}_0)}{R} > \xi, \tag{10}$$

we drop the sample points on the circle, where $0 < \xi < 1$ is the limit or threshold value, R is the radius of minimum circum circle.

When there are outlier in the samples, the center of minimum circum circle will deviate from the center of gravity. The distribution of samples can not be described well by this circle. The accuracy of classification will also decrease. Based on the above algorithm, the outliers on the minimum circum circle can be deleted. The performance of the classifier will be improved.

5.3 Improved Algorithm 3

The linear classifier cannot solve the nonlinear problem. As a result, we adopt the combined classifier based on subclasses to solve this problem.

First, the samples of each class was divided into some subclasses by the clustering algorithm such as k-means. Then we build one subclass classifier for each two subclasses belong to the different classes as showed in Fig. 3. The final classifier is combined by these subclass classifiers. When we discriminate an object, each subclass classifier can give a discriminant class. Then we will determine the final class of the object by the voting of all the subclass classifiers. The class obtains the most votes will be the class of the object.

In Fig. 3, the samples of two classes is linear unseparable. Then these samples are divided into four subclasses, and four subclass classifiers are built. Based on the combination of the subclass classifiers, the classification problem can be solved accurately.

6 Experiments and Results Analysis

In the experiments, we adopted 9 UCI machine learning datasets[1] to train and test the performance of our classifier. The description of all the datasets was listed in the table 2, which showed the number of samples, features and classes for each dataset.

The original dataset was randomly partitioned into 2 subsamples. One subsamples was used for testing, and the other one was used as training data. The classifiers would be estimated by their accuracy on the test set.

When the number of class in a dateset was over two, we changed the multiclass problem into multiple binary classification problems. We build a series of binary classifiers which distinguish between every pair of classes (one-versus-one). When there was a new sample to discriminate, every binary classifier would assign the sample to one of the two classes. Then the vote for this class was increased by one. Finally, the class with the most votes would be the class of this new sample. We compared the following classifiers in our experiment.

- CMCC: original classifier based on minimum circum circle.
- CMCC1: improved algorithm 1 of CMCC.
- CMCC2: improved algorithm 2 of CMCC.
- CMCC3: improved algorithm 2 of CMCC.
- ICMCC: improved CMCC based on algorithm 1, 2 and 3.
- LDA: linear discriminant analysis.
- SVM: support vector machine.
- LDAS: LDA based on subclass classifier.
- SVMS: linear SVM based on subclass classifier.

First, we compared the CMCC with other linear classifiers including linear SVM (SVM without kernel), LDA, CMCC1 and CMCC2. The experimental results on all the datasets were shown in the table 1.

Table 1. Accuracy comparison of the linear classifiers on data sets

Accuracy	linear SVM	LDA	CMCC	CMCC1	CMCC2
Iris	96.00%	96.00%	92.00%	**97.33%**	96.00%
Haberman	62.74%	**65.35%**	60.13%	64.05%	61.43%
Vote	93.10%	93.10%	91.38%	**93.97%**	91.38%
Dermatology	93.99%	94.54%	93.44%	**95.63%**	93.44%
Cancer	95.96%	95.96%	92.93%	95.96%	94.95%
Heart	**79.05%**	78.38%	75.67%	78.38%	78.38%
Sonar	58.65%	57.69%	56.73%	**59.62%**	57.69%
Teach	52.00%	52.00%	50.67%	**53.33%**	52.00%
Wine	**97.75%**	96.63%	96.63%	**97.75%**	96.63%

[1] The datasets can be found in `http://archive.ics.uci.edu/ml/datasets.html`

Table 2. Description of all the datasets

Data Set	Samples	Features	Classes
Iris	150	4	3
Haberman	306	3	2
Vote	232	16	2
Dermatology	366	33	6
Cancer	198	32	2
Heart	297	13	5
Sonar	208	60	2
Teach	151	3	3
Wine	178	13	3

Table 3. Accuracy of combined classifiers on data sets

Accuracy	SVMS	LDAS	CMCC3
Iris	96.00%	96.00%	96.00%
Haberman	71.24%	71.90%	**73.85%**
Vote	**94.83%**	93.97%	93.10%
Dermatology	93.44%	**94.54%**	93.99%
Cancer	94.95%	94.95%	**95.96%**
Heart	77.03%	78.38%	**79.05%**
Sonar	69.23%	**71.15%**	70.19%
Teach	59.21%	**61.33%**	59.21%
Wine	97.75%	97.75%	97.75%

From the table, the accuracy of CMCC1 is higher than CMCC on all the datasets and the accuracy of CMCC2 is not lower than CMCC. That proves the proposed algorithms indeed improve the performance of the CMCC. The CMCC1 increases the freedom degree of the CMCC. As a result, the accuracy of CMCC1 is raised. Due to removing of the outliers, the performance of CMCC2 is better. However, the improvement of CMCC2 is not so significant as CMCC1. That means the increasing of the freedom degree is more important for our classifier.

Compared with the LDA and linear SVM, the accuracy of CMCC1 is higher except on the datasets of Haberman and Heart, which shows the effectiveness of CMCC1. On the most datasets, the linear SVM is not better than LDA or CMCC1. That means the SVM without kernel has not enough ability to discriminate the object accurately.

Then we compared the combined classifiers. In the table 3, the accuracy of combined classifiers on all the datasets was presented, including SVMS, LDAS and CMCC3.

From the table, the accuracy of LDAS and CMCC3 are both higher than SVMS except on the dataset of Vote. That proves the effectiveness of LDAS and CMCC3. The performance of LDAS is similar to CMCC3. The accuracy of LDAS is higher on Dermatology, Sonar and Teach and the accuracy of CMCC3 is higher on Haberman, Cancer and Heart. That means, when the subclass is small, the the performances of LDA and CMCC are close.

Besides, we compared CMCC3 with CMCC in table 1. The accuracy of CMCC3 is significantly higher than CMCC on all the datasets. That means the performance of CMCC is improved significantly by CMCC3 and the nonlinear classifier is usually more effective on these datasets.

At last, we compared performance of all the nonlinear classifiers. In the table 4, the accuracies of SVM (with kernel), LDAS, CMCC3 and ICMCC were shown. From the table, the ICMCC obtains the the highest classification accuracy except on Dermatology, Heart and Sonar. The effectiveness of our classifier can be proved. The performance of SVM is close to ICMCC. It gives the highest accuracy on Dermatology, Heart and Sonar. Compared with the linear SVM in table 1 and SVMS in table 3, The importance of kernel function for SVM is obvious.

Table 4. Accuracy of nonlinear classifiers on data sets

Accuracy	SVM	LDAS	CMCC3	ICMCC
Iris	96.00%	96.00%	96.00%	**97.33%**
Haberman	76.47%	71.90%	73.85%	**77.12%**
Vote	95.69%	93.97%	93.10%	95.69%
Dermatology	**96.72%**	94.54%	93.99%	95.63%
Cancer	95.96%	94.95%	95.96%	**96.97%**
Heart	**87.16%**	78.38%	79.05%	83.78%
Sonar	**75.00%**	71.15%	70.19%	74.04%
Teach	62.67%	61.33%	59.21%	**64.00%**
Wine	97.75%	97.75%	97.75%	**98.88%**

We also compared the time cost of these classifiers, which were presented in the table 5. From the table, LDAS is the fastest due to its small amount of computation. The time cost of SVM is higher than LDAS and lower than CMCC3 and ICMCC. However, when the dimension is high, the ICMCC and CMCC3 can be faster than SVM. For example, the time cost of CMCC3 is lowest on Sonar and Dermatology. The ICMCC is also faster on Sonar. That proves our classifier is more effective for high dimensional data.

Over all, the improved CMCC is an effective classifier and gives a good performance on all the datasets.

Table 5. Time cost of nonlinear classifiers on data sets

Time	SVM	LDAS	CMCC3	ICMCC
Iris	0.1755	**0.1074**	0.2669	0.3566
Haberman	0.4023	**0.3671**	1.8242	2.5231
Vote	2.8757	2.1896	**2.0307**	3.0364
Dermatology	4.3756	**3.3358**	3.9489	4.3808
Cancer	5.1666	**4.0829**	6.1187	8.1171
Heart	1.3782	**0.7444**	1.5763	2.0315
Sonar	1.8208	1.9296	**1.7258**	1.8018
Teach	0.249	**0.1022**	0.3675	0.4248
Wine	0.6398	**0.3666**	1.3718	1.5807

7 Conclusion

In this paper, we proposed a classifier based on minimum circum circle. As a new linear classifier, it calculates the minimum circum circle for the samples of each class. Then the linear classifier is determined by the center of the minimum circum circle. The connecting line of the centers will be perpendicular to the separating hyperplane and the linear classifier who gives the highest accuracy will be the final classifier.

Based on the original CMCC, we proposed three improved algorithms. The improved CMCC increases the freedom degree of the classifier, deletes the outliers in the training samples and solves the linear unseparable problem. The performance of CMCC has been improved based on these algorithms.

In the experiments, we compared CMCC and its improved algorithms with other classifiers. From the experimental results, CMCC showed a relatively high classification accuracy. The proposed improved algorithms improved the performance of CMCC effectively. The effectiveness of the CMCC can be proved.

In the future work, an algorithm to calculate the minimum circum circle in higher dimension will be discussed and the kernel method will be introduced in our classifier.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (NSFC), under grant number 61170057 and 60875080.

References

1. Chapelle, O., Scholkopf, B., Zien, A.: Semi-Supervised Learning. MIT Press, Cambridge (2006)
2. Torkkola, K.: Linear Discriminant Analysis in Document Classification. In: IEEE ICDM Workshop on Text Mining (2001)
3. Osuna, E., Freund, R., Girosit, F.: Training support vector machines: an application to face detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 130–136 (1997)
4. Kolter, J.Z., Maloof, M.A.: Learning to Detect and Classify Malicious Executables in the Wild. The Journal of Machine Learning Research 7, 2721–2744 (2006)
5. Ng, A.Y., Jordan, M.I.: On Discriminative vs. Generative Classifiers: A comparison of logistic regression and Naive Bayes. Neural Information Processing Systems 2(14), 841–848 (2002)
6. Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Mullers, K.R.: Fisher discriminant analysis with kernels. In: Neural Networks for Signal Processing IX, pp. 41–48 (1999)
7. Domingos, P., Pazzani, M.: On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning 29, 103–137 (1997)
8. Agresti, A.: Building and applying logistic regression models. An Introduction to Categorical Data Analysis, p. 138. Wiley, Hoboken (2007)
9. Gallant, S.I.: Perceptron-based learning algorithms. IEEE Transactions on Neural Networks 1(2), 179–191 (1990)
10. Cristianini, N., Taylor, J.S.: An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press (2000)