Local Bayesian Based Rejection Method for HSC Ensemble

Qing He[1], Wenjuan Luo[1,2], Fuzhen Zhuang[1,2], and Zhongzhi Shi[1]

[1] The Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, China
[2] Graduate University of Chinese Academy of Sciences, Beijing 100049, China
{heq,luowj,zhuangfz,shizz}@ics.ict.ac.cn

Abstract. Based on Jordan Curve Theorem, a universal classification method, called Hyper Surface Classifier (HSC) was proposed in 2002. Experiments showed the efficiency and effectiveness of this algorithm. Afterwards, an ensemble manner for HSC(HSC Ensemble), which generates sub classifiers with every 3 dimensions of data, has been proposed to deal with high dimensional datasets. However, as a kind of covering algorithm, HSC Ensemble also suffers from rejection which is a common problem in covering algorithms. In this paper, we propose a local bayesian based rejection method(LBBR) to deal with the rejection problem in HSC Ensemble. Experimental results show that this method can significantly reduce the rejection rate of HSC Ensemble as well as enlarge the coverage of HSC. As a result, even for datasets of high rejection rate more than 80%, this method can still achieve good performance.

Keywords: HyperSurface Classification (HSC); HSC Ensemble; Rejection.

1 Introduction

Among various kinds of classification algorithms in machine learning, there exists a family of covering algorithms (also called rule learning algorithms) which follow the so-called separate-and-conquer or covering strategy. In detail, these methods learn a rule at a time, which explains(covers) a part of the training examples, then the covered examples are removed and the rest examples are used to learn new rules successively [1]. This procedure is carried on until all the training examples are covered. While in the classifying phase, for a test example, rules are tried until it satisfies any of the learned rules and then it is classified as the label implied by the satisfied rule.

Different covering algorithms differ in the way how single rules are generated. Based on the McCulloch-Pitts Neural Model, Zhang et al. [2] proposed a covering algorithm for classification which models an M-P neuron as a covering on the input space and constructs a set of labeled sphere neighborhoods for classification. Based on the same model, Wu et al. [3] combined the kernel function algorithm of SVM and spherical domain covering algorithm of constructive machine learning method, and made improvements for the algorithm.

L. Zhang, J. Kwok, and B.-L. Lu (Eds.): ISNN 2010, Part I, LNCS 6063, pp. 404–412, 2010.

Based on Jordan Curve Theorem, He et al. [4] proposed another covering algorithm for classification which constructs hypersurface homeomorphic to lower dimensional sphere of the input space, and class labels are obtained according to whether the intersecting number between the test sample and the hypersurface is odd or even. In fact, He et al. [4] view the hypersurface as single rules and generate them recursively, while Zhang et al. [2] and Wu et al. [3] consider the labeled sphere neighborhoods to be single rules and construct them iteratively.

In machine learning and data mining, a typical assumption is that the training data and the test data are drawn from the same feature space and follow the same distribution [5]. However, in the real world, distributions of training data and test data do not actually match, and then rejection happens.

There are mainly two types of rejection, one rises when a new class label appears only in the test data [6], and the other rises typically in covering algorithms when test samples belong to the areas which no classifier(or covering) covers [3]. In case of the former kind of rejection, there are several strategies to handle, such as *distance-based reject-option* [7], *ambiguity reject-option* [7] and *combinations of one-class and supervised classifiers* [8].

However, the classification threshold or the rejection threshold for the above mentioned methods are difficult to choose [6].

In case of the latter kind of rejection, Zhang et al. [9] put forward a probabilistic model which utilizes a Gaussian kernel covering function, they also implement Expectation Maximization Algorithm for global optimization. As a result, this model broadens the application domain of their covering algorithm and reduces the rejection rate. However, it remains a difficult problem on how to select a proper kernel function.

As a kind of covering algorithm also suffering from rejection, HSC was proposed to classify spirals in [4]. In case of real world data, He et al. [10] proposed a dimension reduction method for HSC which transforms high dimensional dataset into three-dimensional dataset. Afterwards, for high dimensional classification, Zhao et al. [11] proposed HSC Ensemble, which divides a dataset vertically into sub datasets with every three dimensions of data, and then ensembles sub classifiers generated by every sub dataset. However, Zhao et al. [11] did not take rejection into consideration. In this paper, we study the rejection problem of HSC Ensemble and only discuss the second type of rejection, i.e., only rejection when test examples exceed the boundary of HSC is investigated. The rest of this paper is organized as follows: in Section 2, we outline the main idea of hyper surface classifier (HSC). Then in Section 3, we focus on HSC Ensemble and the rejection method, the results of which are presented in Section 4, and Section 5 concludes our paper.

2 Main Idea of HSC Algorithm

HSC is a universal classification method based on Jordan Curve Theorem in topology. Compared to SVM, this approach can directly solve the nonlinear classification problem in the original space other than higher dimensional space, therefore without the use of kernel function.

Jordan Curve Theorem. Let X be a closed set in n-dimensional space . If X is homeomorphic to a sphere in n-dimensional space, then its complement has two connected components, one called inside, the other called outside. Based on the Jordan Curve Theorem, n-dimensional space can be separated by a double-sided surface that is homeomorphic to a sphere in $(n-1)$-dimensional sphere. So X can be seen as a separating hyper surface.

Classification Theorem. For any given point $x \in R^n/X$, x is inside of X, iff, the intersecting number between any radial from x and X is odd; and x is outside of X, iff, the above-mentioned intersecting number is even.

Based on the above theorems, we can construct hypersurface and use it for classification. Main steps are in the following.

Main Steps of HSC. There are two major procedures in HSC, one is the training procedure, and the other is the testing procedure.

Training Procedure

Step 1. Input the training samples, containing k categories and d-dimensions. Let the training samples be distributed within a rectangular region.

Step 2. Divide the region into $\overbrace{10 \times 10 \times \cdots \times 10}(10^d)$ small regions called units. Here, 10 is the number of divisions we perform on each dimension practically, and actually it could be any number that is above 1.

Step 3. If there are some units containing samples from two or more different categories, then divide them into smaller units recursive until each unit covers at most samples from the same category.

Step 4. Label each unit with $1, 2, \cdots, k$ according to the category of the samples inside, and unite the adjacent units with the same labels into a bigger unit.

Step 5. For each unit, save its contour as a link, and this represents a piece of hyper surface.

Testing Procedure

In the testing procedure, we just count the number of intersections between the radial starting from the test sample and the trained hypersurface. If the number is odd, then the sample is marked as the class indicated by the link.

The classification algorithm based on hypersurface is a polynomial algorithm if the same class samples are distributed in finite connected components. Experiments showed that this method can work fairly well in both accuracy and efficiency in three-dimensional space even for large size data up to 10^7. Specifically, [12] pointed out that models trained from Minimal Consistent Subset can correctly classify all the remaining points in the sample set.

3 HSC Ensemble and Rejection Method

3.1 HSC Ensemble

By attaching equal importance to each feature, HSC Ensemble firstly groups the overall dataset into sub datasets with every three dimension of data, when there

is less than three dimension for the last sub dataset, one dimension or two from the last but one sub dataset is reused. Detailed algorithms can be found in [11].

3.2 Rejection Problem

Fig. 1 illustrates the rejection problem in two dimensional space, where the hypersurface(also called HSC Classifier or HSC model in this paper) is constructed from combinations of rectangles. Suppose there are two classes in the figure, one is covered by the hypersurface on the bottom left corner (depicted with "+"), and the other is covered by the hypersurface on the top right corner (depicted with "."). When test data are located in the covering rectangles, there is no rejection. However, when test data come from the ellipse region, which is not covered by any of the hypersurface, all the data points in this region become unrecognized and rejection happens.

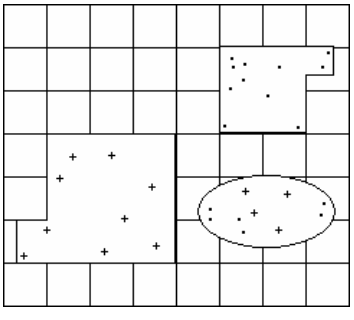

Fig. 1. Rejection Problem in 2 Dimension

3.3 Local Bayesian Based Rejection Method

HSC divides the feature space recursively until all pure covers(hypersurface) are constructed. The way that HSC covers feature space is so conservative that a cover only covers points exactly from the same class. Moreover, from the main steps of HSC, we can see that, after each dividing, the edge length of HSC decreases to 1/10 of the original edge length before dividing.

As a result, the more deeply HSC divides the feature space, the shorter the edge of HSC would be and the more unrecognized fragments there would be in the feature space. In an extreme case, each HSC cover may cover only 1 point with extremely short edges. Thus when class labels are assigned to points in a haphazard manner, the feature space has to be divided recursively in order to get pure HSC covers. Consequently, small fractions of attribute range are rejected. In case of high dimensionality, the coverage of HSC may shrink exponentially when too many fractions of attribute range are rejected.

From the above analysis, there are two strict constraints restricting the coverage of HSC. The first is that each piece of hypersurface should only cover points

that have determinate class labels the same as the hypersurface. The second is that edge length of hypersurface becomes smaller during recursive dividing.

Encouraged by the above analysis, we proposed the following method to deal with rejection problem in HSC. From the bayesian inference theory we have:

$$P(C|X) = \frac{P(C,X)}{P(X)} = \frac{P(X|C) \times P(C)}{P(X)} \tag{1}$$

In Equation (1), X denotes a sample, $P(X)$ is the probability of X, while C is a class label and $P(C)$ corresponds to its probability. $P(C|X)$ denotes the probability of class label C conditioned on X and $P(X|C)$ is the probability of X conditioned on C. For a classification problem, the goal is to find the class label C which has the highest $P(C|X)$. As $P(X)$ is the same for any class label, so the classification problem is to estimate the probability $P(X|C) \times P(C)$, as to find the class label C with the maximal $P(C,X)$.

However, when dealing with rejection, it becomes unrealistic to estimate $P(C,X)$ as there is no hypersurface covering X. In order to solve this problem, we choose points from the local area of X to estimate the probability $P(C,X)$. Moreover, instead of viewing X from 3-dimensional perspective as from HSC, we analyze X from each-dimensional perspective and then ensemble the probability $P(C,X)$ for all possible C.

Here, suppose the dimension number of X is d, and X is represented as $X(x_1, x_2, x_3, ..., x_d)$, we define the local area of X to be:

$$Local\_Area(X) = \{P(p_1, ..., p_d) | P \in TS, \exists i \in (1, ..., d), (\lfloor x_i \rfloor \leq p_i < \lceil x_i \rceil)\} \tag{2}$$

In Equation (2), $P(p_1, ..., p_d)$ is a point from training set (denoted as TS), while $\lfloor x_i \rfloor$ denotes the largest integer smaller than x_i and $\lceil x_i \rceil$ is the smallest integer larger than x_i. From the equation above, we can see that for a point which has at least one dimension with attribute value p_i falling into the scope of $\lfloor x_i \rfloor$ and $\lceil x_i \rceil$, it will be included in $Local\_Area(X)$. This is different from the cover of hypersurface which follows:

$$Cover(H) = \{P(p_1, ..., p_d) | P \in TS, \forall i \in (1, ..., d), (h_{i_{low}} \leq p_i < h_{i_{up}})\} \tag{3}$$

In Equation (3), P is the same as in Equation (2), while $H([h_{1_{low}}, h_{1_{up}}), [h_{2_{low}}, h_{2_{up}}), ..., [h_{d_{low}}, h_{d_{up}}))$ stands for a piece of hypersurface, where $[h_{i_{low}}, h_{i_{up}})$ composes the edge of H in the i_{th} dimension. Compare Equation (2) with Equation (3), we can see that the local area has a much more smooth boundary, while an HSC Classifier has a so strict boundary that many points would be rejected. Moreover, the edge length of local area remains 1, however, based on the dividing mechanism of HSC, the edge length of HSC is ≤ 1 ("=" happens only in the first dividing of feature space). In consequence, $Local\_Area(X)$ has a much larger coverage than $Cover(H)$. Take a 2-dimensional example, a point $A(2.1, 2.2)$ corresponding to a hypersurface of $H : \{X(x,y) | 2 \leq x < 3, 2 \leq y < 3\}$ and a local area $L : \{X(x,y) | 2 \leq x < 3\} \bigcup \{X(x,y) | 2 \leq y < 3\}$, another point $B(2.9, 3.1)$ would not belong to the hypersurface H but fall into L.

Since we only aim to maximize $P(C,X)$ in the $Local\_Area(X)$, therefore, when we have the $Local\_Area(X)$ constructed, calculating the probability of

$P(C, X)$ in $Local\_Area(X)$ is equal to calculating the probability of $P(C)$ in $Local\_Area(X)$, as all X are from the same $Local\_Area(X)$. For a test example X that is rejected, detailed operations for classification are as follows:

Step 1. Scan the training data and construct $Local\_Area(X)$ defined in Equation (2).
Step 2. For each class label C, for all the points in $Local\_Area(X)$, compute $P(C)$.
Step 3. Select the class label C which has the highest $P(C)$ and predict X as label C.

In step 2, we compute $P(C)$ in the $Local\_Area(X)$, as mentioned above, maximizing this $P(C)$ is equivalent to maximizing the $P(C, X)$ in $Local\_Area(X)$. And we call the above algorithm as local bayesian based rejection method(LBBR). On the time complexity of the above algorithm, suppose N is the number of training instances, d is the number of dimensions and c is the number of total class labels. The average-run-time of Step 1 is $O(d * N)$. Since we use a hash tree to store class labels, the average-run-time of Step 2 is $O(\alpha * d * N)$, where α is the percentage of points in the $Local\_Area$ to the whole dataset, so $\alpha < 1$. For Step 3, the average-run-time is $O(c)$. As c is much smaller than $d * N$, thus the overall average time is $O(d * N + \alpha * d * N) = O(\beta * d * N)$, with $\beta < 2$.

Therefore, the time complexity of local bayesian based rejection(LBBR) method is linear with the size of training data. Thus LBBR has a good scalability. Experimental results are given in Section 4. Also, we make use of the ROC (Receiver Operating Characteristic) space for a clear comparison between HSC with LBBR (denoted as "HSC+LBBR" in the following) and HSC without.

4 Experiments

4.1 Experimental Data

We use data sets from UCI repository [13] to evaluate our rejection method. Table 1 gives the detailed description of data sets adopted. We divide training sets and testing sets in the same way as [11], i.e., we randomly choose about 2/3 of data for training and the rest for testing. For some of the datasets in Table 1, we delete some ineffective or redundant attributes. Take the dataset image for example, we delete 2 attributes which are not important for classification but actually impede the construction of sub HSC Classifiers.

Table 1. Description of data Sets

Data Set	Number of Dimensions	Number Of training Samples	Number Of testing Samples	Number Of classes
hayes	5	80	71	3
breast	9	369	200	2
glass	9	151	54	7
image	17	150	60	7
parkins	21	130	65	2
inosphere	34	250	101	2
libras	90	300	59	15

4.2 Experimental Results

We implemented LBBR and results are shown in Table 2. In Table 2, Recall denotes the accuracy of HSC on training data and HSC+LBBR stands for the accuracy of "HSC+LBBR" on test data. Accuracy of HSC denotes, for all the test samples that are **not** rejected, the accuracy of HSC. And Accuracy of LBBR is, for all the rejected test samples, the accuracy of LBBR. From Rejection Rate column, we can see that different datasets have different rejection rates. Besides, we can see from the table that when rejection rate is high, the overall accuracy mainly depends on the accuracy of LBBR. Thus when local Bayesian performs well, such as on the dataset libras, the overall accuracy would be high.

Table 2. Results of HSC Ensemble on Rejection Problem

Data Set	Recall	Rejection Rate	Accuracy of HSC	Accuracy of LBBR	HSC+LBBR
hayes	86.25%	50.7%	88.57%	72.22%	80.28%
breast	99.46%	24.85%	99.99%	96.34%	99.09%
glass	100.00%	98.14%	100.00%	66.04%	66.67%
image	100.00%	100.00%	-	85.00%	85.00%
parkins	100.00%	100.00%	-	76.92%	76.92%
inosphere	100.00%	95.05%	99.95%	91.67%	92.08%
libras	100.00%	100.00%	-	91.53%	91.53%

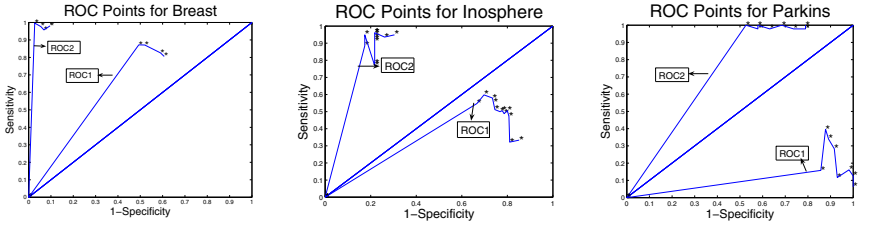

Fig. 2. ROC **Points** of HSC Classifiers With LBBR And Without

Fig. 2 shows the ROC **points** of HSC Classifiers on 3 different datasets, corresponding to Breast, Inosphere and Parkins, respectively. In the ROC space, each point is coordinated as $(1 - specificity, sensitivity)$. For specificity and sensitivity, we have:

$$Specificity = \frac{TN}{TN + FP}, Sensitivity = \frac{TP}{TP + FN} \tag{4}$$

In the above equations, TP denotes the number of True Positives, FN-False Negatives, TN-True Negatives and FP-False Positives. Thus a ROC point describes the classification capability of a classifier. For a binary classification problem, an ideal classifier would obtain a sensitivity of 1 and a specificity of 1, thus, in the ROC space, it would be represented as the point(0,1). Classifiers that take random guess with $Sensitivity = 1 - Specificity$, would be lined as $y = x$. Thus,

classifiers better than random guess should be represented as points above the line($y = x$) in the ROC space.

In all subgraphs of Fig. 2, ROC2 includes the ROC **points** of HSC+LBBR classifiers, and ROC1 includes the ROC **points** of HSC classifiers. All points on ROC2 are closer to (0,1) than points on ROC1, which indicates that sub HSC+LBBR classifiers outperform sub HSC classifiers. The larger is the area between ROC2 and ROC1, the more effective is LBBR. Looking at the rejection rate of breast, inosphere and parkins in Table 2, we can see that the area between ROC1 and ROC2 becomes larger as the rejection rate grows. From the discussion and comparisons above, we can see that LBBR for HSC is effective.

5 Conclusion

In this paper, after analyzing the cause of rejection in HSC Ensemble, we propose a local bayesian based rejection method(LBBR) to solve the rejection problem in HSC Ensemble. This method computes the probability of a test sample belonging to a certain class in the local area of the test sample and get the class label of test sample in use of Bayesian method. As a result, this method not only smooths the edge of hypersurface, but also enlarges the coverage of HSC Ensemble. As shown by our experiments, this method obtains high accuracy as well as obvious improvement for HSC classifiers. In conclusion, the local bayesian based method could deal with the rejection problem of HSC Ensemble effectively.

Acknowledgements

This work is supported by the National Science Foundation of China (No.60675010, 60933004, 60975039), 863 National High-Tech Program (No.2007AA01Z132), National Basic Research Priorities Programme (No.2007CB311004) and National Science and Technology Support Plan (No.2006BAC08B06).

References

1. Furnkranz, J.: ROC n Rule Learning Towards a Better Understanding of Covering Algorithms. Machine Learning 58, 39–77 (2005)
2. Zhang, L., Zhang, B.: A Geometrical Representation of McCullochCPitts Neural Model and Its Applications. IEEE Transactions on Neural Networks 10(4) (1999)
3. Wu, T., Zhang, L., Yan-Ping, Z.: Kernel Covering Algorithm for Machine Learning. Chinese Journal of Computers 28(8) (2005)
4. He, Q., Shi, Z.-Z., Ren, L.-A., Lee, E.S.: A Novel Classification Method Based on Hyper Surface. International Journal of Mathematical and Computer Modeling, 395–407 (2003)
5. Pan, S.J., Yang, Q.: A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering (October 12, 2009)
6. Landgrebe, T.C.W., Tax, D.M.J., Paclk, P., Duin, R.P.W.: The interaction between classification and reject performance for distance-based reject-option classifiers. Pattern Recognition Letters 27(8), 908–917 (2006)

7. Dubuisson, B., Masson, M.: A statistical decision rule with incomplete knowledge about classes. Pattern Recognition 26(1), 155–165 (1993)
8. Landgrebe, T., Tax, D., Paclk, P., Duin, R., Andrew, C.: A combining strategy for ill-defined problems. In: Fifteenth Ann. Sympos. of the Pattern Recognition Association of South Africa, pp. 57–62 (2004)
9. Zhang, L., Wu, T., Zhou, Y., Zhang, Y.P.: Probabilistic Model for Covering Algorithm. Journal of Software 18(11), 2691–2699 (2007)
10. He, Q., Zhao, X., Shi, Z.: Classification based on dimension transposition for high dimension dataInternational Journal Soft Computing-A Fusion of Foundations. Methodologies and Applications, 329–334 (2006)
11. Zhao, X.R., He, Q., Shi, Z.Z.: HyperSurface Classifiers Ensemble for High Dimensional Data sets. In: Wang, J., Yi, Z., Żurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3971, pp. 1299–1304. Springer, Heidelberg (2006)
12. He, Q., Zhao, X.-R., Shi, Z.-Z.: Minimal consistent subset for hyper surface classification method. International Journal of Pattern Recognition and Artificial Iintelligence 22(1) (2008)
13. `http://archive.ics.uci.edu/ml/`