OPTIMIZATION OF BAGGING CLASSIFIERS BASED ON SBCB ALGORITHM

XIAO-DONG ZENG, SAM CHAO, FAI WONG

Faculty of Science and Technology, University of Macau, Macau, China
E-MAIL: ma96506@umac.mo, lidiasc@umac.mo, derekfw@umac.mo

Abstract:

Bagging (Bootstrap Aggregating) has been proved to be a useful, effective and simple ensemble learning methodology. In generic bagging methods, all the classifiers which are trained on the different training datasets created by bootstrap resampling original datasets would be seen as base classifiers and their results would be combined to compute final result. This paper proposed a novel ensemble model that refines the bagging algorithm with an optimization process. The optimization process mainly emphasizes on how to select the optimal classifiers according to the accuracy and diversity of the base classifiers. While the select classifiers constitute the final base classifiers. The empirical results reveal that the new model does outperform the original method in terms of learning accuracy and complexity.

Keywords:

Bagging; Classifier optimization; Ensemble learning; Selective ensemble

1. Introduction

Ensemble learning employs multiple learners and combines their prediction capabilities [1]. By combining classifiers, we are aiming at a more accurate classification decision at the expense of increased complexity. According to [2], there are three types of reasons (statistical, computational and representational) why a classifier ensemble might be better than a single classifier.

In general, an ensemble model contains four layers: input layer, feature layer, base classifier layer and fusion layer. Input layer mainly focuses on providing various kinds of datasets. Feature layer creates different feature sets using output of input layer. Base classifier layer mainly emphasizes how to create base classifiers and design the structure of classifiers. Fusion layer is in charge of how to combine the prediction results from base classifiers using some fusion rules, ensemble rules, or combine rule .etc.

The representative ensemble methods are Adaboost and Bagging. Adaboost [3], which sequentially generates a series of base learners, where the training instances that are wrongly predicted by a base learner will play more important role in the training of its subsequent learner. Bagging (Bootstrap Aggregating) [4] is not to sequentially generate base learners, but parallel bootstrapping resamples in diffident datasets to create diverse base classifiers.

The general ensemble methods would ensemble all the classifiers generated using training datasets. Although such methods have been made a great performance, recently some researchers like Zhou [5] have begun to prove “many is better than all”, which means that selecting partial classifiers could have the same or even better performance comparing using all classifiers.

This paper proposed an optimization method which selects the optimal classifiers from original classifiers to be the final base classifiers, which operates between fusion layer and base classifier layer. In classifier optimization process, we mainly consider the accuracy and diversity of the classifiers to be selected. We embedded this method into bagging algorithm to build a new model and are seeking to acquire better performance.

The structure of this paper is as follows: section 2 describes the principle of bagging and how to measure the accuracy and diversity of the classifiers. In section 3, we introduce a new algorithm containing the optimization process mentioned above base on bagging. In section 4 we present experimental work on the new algorithm, including the corresponding experimental process and final results. The conclusion would be given in section 5.

2. Related work

2.1. Bagging

Bagging stands for Bootstrap Aggregating, which is one of the most famous and successful ensemble learning methods. Bagging was introduced by Breiman [6]. The main idea of bagging is easy to understand. Bagging wants to parallel create diverse classifiers and then ensemble them, so it selects certain base classifier algorithm to train base classifiers on random redistribution training datasets. According to [6], each training dataset in bagging is generated by randomly drawing with replacement, *N*

978-1-4244-6525-5/10/$26.00 ©2010 IEEE

examples - where N is the size of the original training datasets. Many of the original examples may be repeated in the resulting training datasets while others may be left out. The classifiers trained by training datasets would be regarded as base classifiers. In test phase, input x, it would be predicted by every base classifier and the predictions would be combined by the plurality vote.

2.2. Measuring accuracy and diversity

Classification accuracy is an easy comprehensible paradigm that represents the ratios of examples that a classifier correctly recognizes the class of testing examples. To measure the accuracy of a classifier is also an easy way, and calculate according to formula given in Figure 1.

$$CR=\frac{C}{A}$$

CR – The correct rate;

C – The number of sample recognized correctly;

A – The number of all sample;

Figure 1. Formula to calculate accuracy

The higher the correct rate, the more the classification accuracy it is. However, in ensemble classification, it is not necessary that every base classifier gives a high accuracy. Otherwise, the ensemble of relatively perfect classifiers will not give a better result. In other word, classifiers with high accurate put together may lower the complementarily.

On the other hand, diversity can enhance such complementarily on classifiers. Classifier diversity mainly refers to the diverse classifier outputs. Imaging that if a classifier's output does make errors, we could seek to complement it with another classifier. The diversity is therefore a vital requirement for the success of ensemble learning [7]. In practical, it is difficult to measure classifier diversity. According to [7], there are two main streamline to measure diversity: Pairwise Measures and Nonpairwise Measures. Pairwise Measure is a method which considers a pair of classifiers at a time so that an ensemble of L classifiers will produce $L(L\text{-}1)/2$ pairwise diversity. (The Q statistic, the correlation, the disagreement and the double fault) belong to this streamline; Nonpairwise Measure is a totally different method which considers a group of classifiers. Therefore, it just calculates one time to get one value for the ensemble. Nonpairwise measures have many Variances such as (Entropy Measure E , Kohavi-Wolpert Variance, Measurement of Interrater Agreement k, measure of "difficulty"). This paper selected "Entropy Measure E" [9] in the second streamline to test the diversity of classifier. The calculating formula for Entropy Measure E is illustrated in Figure 2. The higher the E is, the more the diversity the classifiers give.

The Entropy Measure E:

$$E=\frac{1}{N}\frac{2}{L-1}\sum_{j=1}^{N}\min\{(\sum_{i=1}^{L}y_{j,i}),(L-\sum_{i=1}^{L}y_{j,i})\}$$

E – Varies between 0 and 1, where 0 indicates no difference and 1indicates the highest possible diversity.

N – The number of instance;

L – The number of classifier;

$y_{j,i}$ – Classifier output;

Figure 2. The Entropy Measure E

3. SBCB algorithm

SBCB (Selecting Base Classifiers on Bagging) is an optimized method base on bagging. The model of this approach is shown in Figure 3.

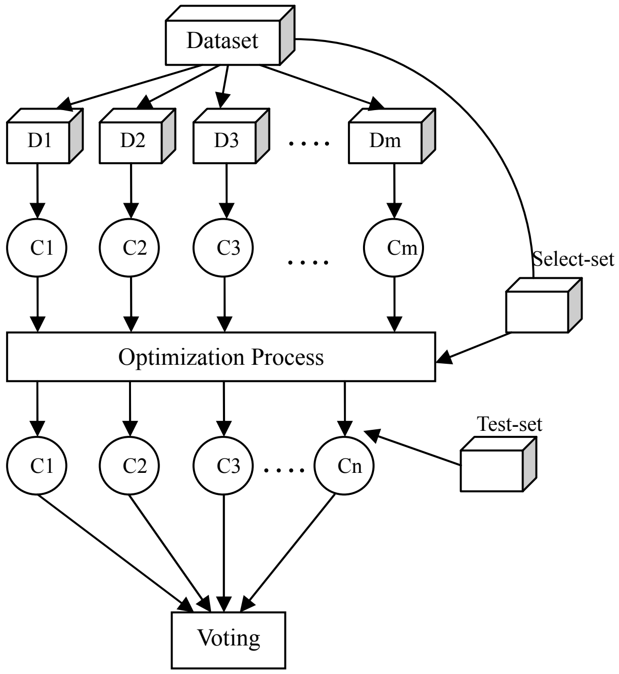

Figure 3. The SBCB model

The ensemble is not directly constructed by original classifiers trained on the bootstrap algorithm that replicates original training set. That means the classifier candidates trained at the beginning of the phase may not be the set of base classifiers that will be evaluated with the test dataset at the second phase. The final base classifiers should be selected through an optimization process. Therefore, the biggest difference between SBCB and the Bagging training paradigm is the step of optimization process, as illustrated in Figure 3. Following section will focus on the description of the optimization process, where how the base classifiers are being selected from the initial set of candidates.

The *optimization process* is the core of the proposed SBCB algorithm. It mainly focuses on how to select classifiers from the original classifier candidates that give a better performance. The selection criterion is similar to selective ensemble idea proposed by Zhou [8]. The hypothesis is that fewer classifiers are not worse than or even better than that of all generated classifiers given by conventional Bagging system. Many research works have illustrated that accuracy and diversity of classifier are two important factors to influent the ensemble performance [7], [9], [11]. Hence, the proposed *optimization process* focuses on picking the set of classifiers to form the base classifiers according to evaluation in terms of accuracy and diversity.

For the evaluation based on accuracy and diversity, general ensemble algorithms tend to consider one of them. But in our optimization process of SBCB, both of these criteria will be considered. However, accuracy and diversity, in some sense, contradict to each other, and we need to find a balance between of these two criteria in probing the base classifiers.

For the accuracy of classifier in ensemble, it would not be defined with high requirement. But this does not mean that accuracy of classifier can be very low. Therefore, in SBCB, classifiers' accurate rate is set to be higher than of 50%. That means the classifier is acceptable when it only slightly correlated with the true classification. And if classifier's accuracy is lower than 0.5, the classifier should be eliminated from the ensemble. This selection metric can help the optimization process to eliminate the classifiers with low classification accuracy, and at the same while, preserve the complementarily between classifiers with high diversity.

The second part of the optimization process is to further filter classifiers based on the measure of diversity, and preserve the classifiers with maximum diversity. The Entropy Measure E mentioned at section 2 is a method that can measure the diversity for all the classifiers together and calculating directly one diversity value for the ensemble. According the following formula, a diversity level of the combination of classifiers could be calculated.

$$E = \frac{1}{N}\frac{2}{L-1}\sum_{j=1}^{N}\min\{(\sum_{i=1}^{L} y_{j,i}),(L-\sum_{i=1}^{L} y_{j,i})\}$$

Therefore, we could apply this measure to create a method to find which classifier has lowest contribution to diversity of set of classifiers and then eliminate it. The selecting diverse classifier process is the follow: The first step calculates the entropy E_0 of the remaining L classifiers after checking accuracy. And then it processes iteration:

```
SBCD
Training phase
1. Initialize the parameters
   • $D$, original dataset;
   • $L$, base classifier learning algorithm;
   • $m$, the number of classifiers to train;
2. For i = 1 to m
   • Take a bootstrap dataset $D_i$ from $D$;
   • Build a classifier $C_i$ using training set $D_i$ and $L$;
   End For;
3. Optimized Process
   • Take a bootstrap dataset as select-set;
   • Let $C_1 \dots C_m$ recognize select-set;
   • Check the accuracy of $C_1$, $C_2 \dots C_m$ using select-set;
     For i = 1 to m
         If ($C_i$.accuracy<0.5) eliminate Classifier $C_i$;
     End for;
     Remaining L classifiers;
   • Select high diverse classifiers
     Calculate $E_0$ of L classifiers;
     While ($E_m < E_0$)
        For i = 1 to L
            Calculate $E_i$;
        End For;
        Find the maximum $E_m$ from $E_1..E_L$;
        If ($E_m > E_0$)
              eliminate the classifier devoting to $E_m$;
              $E_0 = E_m$; L--;
        Else stop;
     Remaining n classifiers;
4. Return an ensemble $C_1$, $C_2 \dots C_n$;
Classification phase
5. let $C_1$, $C_2 \dots C_n$ predict on the input x;
6. The class with the maximum number of votes is
   chosen as the label for x.
```

Figure 4. The SBCB algorithm

take one classifier out and calculate entropy E_i of the rest of classifiers. After this iteration, it should find the maximum E_m in $E_1...E_L$. Maximum E_m represents that taking the corresponding classifier out and the rest of classifiers have most diversity. Then to compare E_m and E_0: If $E_m < E_0$, this selecting diverse classifier process would stop and the final base classifiers are the classifiers with E_0. If $E_m > E_0$, the classifier leading to E_m should be eliminated and let E_0 equal to E_m and continue next iteration until $E_m < E_0$. Please note that $E_m > E_0$ means there exists the set of L-1 classifiers with diversity higher than the one of L classifier. So it needs to remove a classifier from the set. But, if $E_m < E_0$ means that L-1 classifiers cannot give a better result in comparing with L classifiers.

Actually, the optimization process would not select a fixed number of classifiers for every dataset. In other words, for different dataset, the optimization process would select suitable number of classifier by eliminating *bad* classifiers under the conditions of accuracy and diversity.

The whole process of SBCB algorithm is illustrated in Figure 4. Initially, an original dataset and a base classifier learning algorithm are declared. Meanwhile corresponding parameters should be configured. During the training phase, resampling from origin dataset to create *m* training sets is a good way to create diverse classifiers. The next is to train *m* classifiers using training datasets and based on the ensemble training algorithm. Hence, classifiers C_1, C_2.... C_m have been created. Note that, in conventional bagging training algorithm, the process will stop here and the whole training task is completed. But in SBCB, an optimization process as described above would be applied to the candidates of generated classifiers. After optimization process, fewer classifiers combination C_1, C_2, …C_n ($n \leq m$) is selected, and the whole training process is completed. Similar to bagging, the classification phase is to use voting rule to combine the predictions generated from the base classifiers for the test samples.

4. Experimental work

In order to investigate the performance of SBCB algorithm, an empirical comparison between SBCB and Bagging is performed.

The experimental workbench is Weka, a popular suite of machine learning software written in Java, developed at the University of Waikato. In Weka, we can implement new algorithm according to the standard interfaces of Weka.

The experimental datasets are from UCI machine learning repository. In this experiment, we use 14 different datasets by diversity criterion. Datasets are with different sizes and multiple features. Corresponding parameters and distributions on the datasets are shown on Table 1.

TABLE 1. BENCHMARK DATASETS

Dataset	Name	feature	instance	class
D1	anneal	38	901	6
D2	ORIG	38	900	6
D3	audiology	69	225	24
D4	autos	26	205	6
D5	balance-scale	5	629	3
D6	breast-cancer	9	294	2
D7	wisconsin	10	699	2
D8	horse-colic	28	368	2
D9	mushroom	22	8124	2
D10	credit-rating	15	690	2
D11	german_credit	20	1000	2
D12	pima_diabetes	8	768	2
D13	glass	10	214	7
D14	Cleveland	14	303	5

The experiment focuses on the SBCB and Bagging algorithms. It mainly evaluates the accuracy of algorithm. We select two different base classifier algorithms to do the test respectively: REPTree and Naïve Bayes. The original number of classifiers is 10. The evaluating is based on the 4-cross-validation scheme which dataset is divided into four subsets and uses one of them as test dataset to evaluate the performance of the ensemble that trained on the rest of the other three datasets. We would repeat this procedure four times and average the four estimations as the final result. Table 2 shows the empirical results.

TABLE 2. EVALUATIONS

Dataset	Bagging	SBCB	Bagging	SBCB
	REPTree(%)		Naïve bayes(%)	
anneal	97.94	98.13	87.66	90.61
ORIG	91.86	92.14	76.80	78.18
audiology	76.08	73.77	69.82	69.33
autos	60.38	61.24	57.07	57.80
balance-scale	81.85	81.86	89.54	89.42
breast-cancer	68.84	67.71	73.03	73.31
wisconsin	95.13	95.29	96.15	96.09
horse-colic	83.48	83.56	79.18	79.27
mushroom	99.00	99.99	95.63	95.59
credit-rating	86.01	86.61	77.98	78.04
german_credit	73.47	73.00	74.69	74.71
pima_diabetes	76.91	76.95	75.79	75.89

glass	69.90	70.71	70.54	71.23
Cleveland	82.33	82.31	81.20	82.46

From the experiment result in Table 2, we found that SBCB algorithm gives a better performance than Bagging algorithm in most datasets. When the base classifier algorithm is REPTree, correct rates of SBCB on 10 dataset are higher than that of the Bagging algorithm. When the base classifier algorithm is Naïve Bayes, correction rates of SBCB on 9 datasets are improved. That means for performance, we could acquire the same or even better level.

Table 3 shows the number of final base classifiers on different datasets. For Bagging, the number of base classifiers on each dataset is the same. Because bagging algorithm just utilize all the classifiers trained at the beginning of phase to be the final base classifiers on every case. And the number of classifiers which should be trained is predefined. However, as for SBCB, the numbers of base classifier on each datasets are totally different. That means that the optimization process in SBCB has selected suitable number of classifiers to be the final base classifiers according to different condition on different datasets. From another perspective, we could utilize far less classifiers in comparing to bagging, to acquire the same or even better classification result.

TABLE 3. NUMBER OF FINAL BASE CLASSIFIER

Dataset	Bagging	SBCB	Bagging	SBCB
	Number of BC* (REPTree)		Number of BC (Naïve bayes)	
anneal	10	4	10	5
ORIG	10	4	10	8
audiology	10	6	10	6
autos	10	8	10	6
balance-scale	10	9	10	6
breast-cancer	10	6	10	8
wisconsin	10	4	10	4
horse-colic	10	4	10	8
mushroom	10	8	10	6
credit-rating	10	6	10	6
german_credit	10	8	10	8
pima_diabetes	10	8	10	8
glass	10	8	10	6
Cleveland	10	4	10	4

*** Note that BC shorts for base classifier**

5. Conclusions

In this paper, we seek to do some improvement based on the generic bagging algorithm. For this purpose, classifier selection or classifier optimization is a good way to achieve that goal. Therefore, we add an optimization process into the bagging algorithm and hence proposed the SBCB algorithm. The optimization process mainly focuses on selecting better classifiers which are relatively accurate and diverse. From the experiment results, it proves that the optimized model of SBCB does give a better performance comparing with generic bagging on the same datasets.

The further work of our research is to do another testing for this algorithm on more different datasets and using other base classification algorithms such as neural networks and lazy learning. Meanwhile we would like to extend the optimization process as described in this paper to other ensemble leaning model to further verify the performance of the proposed SBCB algorithm.

Acknowledgements

The authors are grateful to Science and Technology Development Fund of Macau for the funding supports of our research, which under the reference No. 017/2009/A.

References

[1] Martine Sewell, “Ensemble Learning”, from http://machine-learning.martinsewell.com/ensembles/ensemble-learning.pdf , April 2007.

[2] Thomas G. Dietterich. “Ensemble learning”, The Handbook of Brain Theory and Neural Networks, Second Edition, 2002.

[3] Yoav Freund, Robert E. Schapire, “A decision -theoretic generalization of on-line learning and an application to boosting”, Proceedings of the 2nd European Conference on Computational Learning Theory, pp. 23-37, 1995.

[4] Leo Breiman, “Bagging predictors”, Machine Learning , Vol 24, No. 2, pp.123–140, 1996.

[5] Zhou Zhihua, Wu Jianxin, Tang Wei, “Ensembling neural networks: many could be better than all”, Artificial Intelligence, Vol 137, pp. 239-263, 2002.

[6] Leo Breiman, “Bagging predictors”, Machine Learning, Vol 24, pp.123-140, 1996.

[7] Ludmila I. Kuncheva, Christopher J. Whitaker, “Measures 0f diversity n classifier ensemble”, Machine Learning, Vol 5l, No. 2, pp. 181-207, 2003.

[8] Geng X, Zhou Zhihua. “Selective ensemble of multiple eigenspaces for face recognition”, Technical Report, AI Lab, Computer Science & Technology Department, Nanjing University, Nanjing, China, Aug. 2003.

[9] Ludmila I. Kuncheva, Christopher J. Whitaker, “Ten measures of diversity in classifier ensembles: limit for two classifiers”, DERA/IEE Workshop on Intelligent Sensor Processing, pp.10/1-10/6, February 2001.

[10] Yoav Freund, Robert E. Schapire, “Experiments with a new boosting algorithm”, Proceedings of the 13th International Conference on Machine Learning, pp. 148-156, 1996.

[11] Pádraig Cunningham and John Carney. “Diversity versus quality in classification ensembles based on feature selection”, Technical Report TCD-CS-2000-02, Department of Computer Science, Trinity College, 2000.

[12] Tin Kam Ho, “Multiple classifier combination: Lessons and the next steps”, Hybrid Methods in Pattern Recognition, pp. 171-198, 2002.

[13] Catherine A. Shipp and Ludmila I. Kuncheva, “Relationships between combination methods and measures of diversity in combining classifiers”, Information Fusion, Vol 3, No. 2, pp. 135-148, 2002.