ATLANTIS PRESS

Advances in Engineering Research, volume 150

4th International Conference on Machinery, Materials and Computer (MACMC 2017)

An improved text classifier based on random forest algorithm - comparative studies on multiple text classifiers

Luo Xin[1,a]

[1]School of Business Administration, South China University of Technology, Guangdong Guangzhou, China

[a]luoxin@scut.edu.cn

Keywords: Natural language processing; Learning algorithm; Random forest; Artificial intelligence

Abstract. Various classifiers have sprung up in recent years. This paper introduces a new intelligent algorithm for text categorization based on improved random forest algorithm. This improvement greatly increases the performance of the original random forest algorithm. The classifier was tested on the reuters-21578 data set and its classification effect was obtained. The classifier is compared with traditional principle similar classifier CART, REPTree and J48. The experimental results show that the classification accuracy of text classifier based on improved random forest algorithm is higher, and it is faster.

Introduction

Due to the increasing amount of text information available on the Internet, interest in automatic analysis of the information has also increased. There are a number of techniques to search, organize, and process this information and one of these techniques is text classification, by which an unknown text document is assigned to one of several classes.

Text classification has been an active topic in computer science for over forty years. There are many different algorithms and techniques used to perform the classification[1]. And, some have proved to be better at certain tasks than others. Naïve Bayes (NB), K-nearest neighbor (k-NN), Support Vector machines (SVM), the Recchio method and Neural Network (NN) have demonstrated good text classification as classic methods.

The classification methods described above are the most commonly used ones in automatic text classification. Each of them has some unique advantages. Some of them are easier to implement, such as k-NN. Others are more complex, but they are more robust and adaptive, such as SVM. There also suffer from additional shortcomings. The linear classifiers, such as Rocchio, may have the cancroids of a class falling outside the clustered documents. To further enhance the performance of automatic text classification, we developed a new method, called RF-Miner, based on Random forests algorithm.

Random forests (also called RF) was introduced by Leo Breiman in 2001[2]. Random Forest is "competitive in accuracy with the best classification algorithms that are out there now". It is a decision-tree-based ensemble classifier that can achieve classification accuracy. Random Forests is wildly applied to many fields ranging from clinical research to financial decision-making[3].

Random Forests does not over fit, runs fast and efficiently on large datasets such. It does not require assumptions on the distribution of the data, which is interesting when different types or scales of input features are used. These outstanding properties make it suitable for text classification.

The paper is organized as follows. In Section 2, the Random Forest classification methodology will be discussed. 3. The classifier base on Random Forest is constructed in Section 3. Experimental results are given in Section 4. Finally, conclusions are drawn in Section 5.

Theory of Random Forests

In this part, the mechanism and some techniques used in Random Forests will be described. The work in this part is a summary from [2]. More detailed discussion can be found in those papers.

Definition of Random Forests. Breiman defines Random Forests:

- A Random Forests is a classifier consisting of a collection of tree-structured classifiers {h(x, Θ_k),k=1,...} where the { Θ_k} are independent identically distributed random vectors and each casts a unit vote for the most popular class at input X.

Random Forests is a multi-classifiers system. It constructs the numerous trees as sub-classifiers (or called internal classifiers). It combines CART's tree generation idea and the bagging predictor to create tree forests. Using the vote from multiple trees, it can yield more accurate, in general, classification on the test data set. Its accuracy is determined by the correlation are weak and the trees are strong, the final vote will be more accurate. The RF is acclaimed as one of the most accuracy classifiers developed to date. Random Forests algorithm is based on **Bagging Sampling** and **CART**. And through a voting system, it becomes a very accurate predictor.

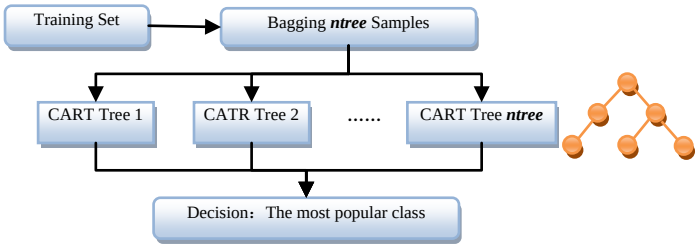

Fig.1, Random Forest work mechanism

Bagging Sampling . Breiman introduces bagging Sampling in 1996. In recent years, it becomes quite popular as other famous sampling methods: boosting (including Adaboosting), v-fold cross-validation, leaf-one cross-validation, randomization, etc. A detailed study and applications of this method are given in [4][5] etc. Bagging is the acronym of **b**ootstrap **agg**regat**ing**. It has following two phrases, sampling and voting. In the above two phrases, the sampling is the kernel and the first randomness in Random Forests.

CART. Classification and Regression tree (also called CART) was introduced by Leo Breiman, et al. In 1984. It creates a tree based on the training data set. At each tree node, it will find the locally best split rule by using some split method.

In Random Forests, each classification tree is a no pruned CART tree. The advantage of using the no pruned tree over using original CART, a pruned tree, is the resulting decrease in the correlation among tress. Even the no pruned tree will affect the strength if each tree but the reduced correlation will improve the final accuracy after combination of all trees. Without pruning, each tree generation will be much simpler and quicker. Therefore, a side benefit of this approach is its assistant on smaller time consumption when generating hundreds of trees.

The classifier based on Random Forests

A text classification model based on random forest algorithm is constructed, and its flow chart is shown in Figure 2.

Step1: Set up text vector set. The text data set is pre processed to form a text VSM set which can be used for random forest algorithm.

Step2: Constructing random forest text classifier.

(1) Use the Bagging method to form *nTree* training sets. Given a training set *D* of size *N*, bagging generates *nTree* new training sets, each of size *N*.

(2) For each training set to generate a CART classification tree with no pruning, the process is as follows:

①Assuming that there are *M* primitive attributes, given a positive integer *mtry*, meet *mtry*<<*M*. Through experiments, set $mtry=M^{1/2}$, when the classification effect is better.

②At each internal node, *mtry* attribute is randomly drawn from the original *M* attribute as a candidate attribute of the split node. In the process of generating the entire forest, the *mtry* remains unchanged.

③The Gini index is used to select the best splitting attribute to split the node from *mtry* candidate attributes.

④Every tree is fully grown, in order to get the maximum tree *Tmax*. Each leaf node of maximum tree is very small, or pure node, or no longer exists attributes can be considered as branches. The node is very small means the number of samples contained within the node is less than a given threshold. Pure point samples belonging to the same class. No pruning the maximum tree *Tmax*.

Step3: Using classifiers. The output of the classifier is determined by majority vote method.

$$c = \text{argmax}_c\left(\frac{1}{ntree}\sum_{k=1}^{ntree} I(h(x,\theta_k)=c)\right) \quad (1)$$

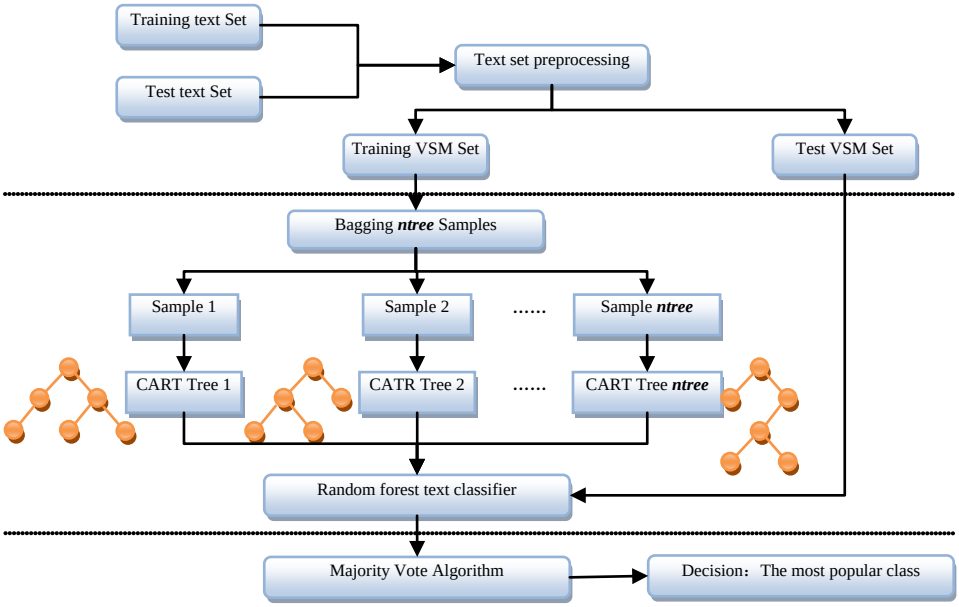

Fig.2, The flow chart of a text classification model based on random forest algorithm

Experiment

Experimental data. The test set **Reuters-21578** is used to evaluate the effect of the classifier based on random forest constructed above. We split the full dataset to training and test sets in ModApte way, get 40 class, 6938 documents for training, 2666 documents for testing. After preprocessing, two VSMs were formed with 200 features.

Experimental parameters. RF classifier has two critical adjustable parameters:

(1)*nodesize*: The number of samples is included in the leaf node.

(2)*ntree*: Number of trees in the forest.

(3)*mtry*: Split variables, the number m of attributes selected at random to choose from when growing each node of a tree.

Normally, when the random forest model is used for classification, *nodesize* takes 1, and when it is used for regression, the 5 is taken as the result, this paper takes *nodesize*=1.

The second step is to select *mtry*. But it is well known that the dependence on this value is not critical. We did a preliminary search over several values of the parameter, from 10 to 20, and found that the variations in performance are minimal. Correspondingly, we choose to keep the default value of m, the square root of M (14 in this case).

After this the RF classifier has only one free parameter, the number of trees. We select it in the ensemble high enough to ensure convergence. Usually 10 to 50 trees (as in our case) are employed. We performed a first experiment to show the potentiality of RF in modeling text data. As explained above, we split the full dataset to training and test sets in ModApte way, containing respectively 6938 and 2666 of the samples. RF models were fitted on the training set, and evaluated after that on the test set. The experiment was repeated 100 times shows that when *ntree*=35 the effect achieves the best.

Another experiment was carried out. We compare this result with three pre-existing and freely available classifiers often used in text classification. They all use If-Then rules, coming from decision tree. First, we model the same sets with CART obtaining an F1-measure over the test sets of 0.754. We also model the training sets with REPTree, the F1-measure over the test sets was 0.705. Then, we model the training sets with J48. With this method the F-measure over the test sets was 0.762, which are similar to RF ones. All results are summarized in Table 1. The experiment shows that RF classifier has the highest F1-Measure, reaching 0.777, while the REPTree classifier's F1-Measure is low, but the time consumption is far lower than other classifiers, only 6.88s.

分类模型	overall time (s)	Precision	Recall	F1-Measure
RF	74.52	0.777	0.792	**0.777**
CART	64.66	0.754	0.775	0.754
REPTree	**6.88**	0.702	0.727	0.705
J48	44.3	0.765	0.775	0.762

Table 1Comparison experiment

Conclusions

Experiments above shows that when nodesize=1， mtry=14， ntree=35,the RF classifier achieved the best results, F1-Measure get 0.777.

We introduce the use of Random Forests as a new and useful modeling technique in the field of Text Classification. This method is easy to use and fast, requiring only the tuning of three parameters (but their value is not critical).Free software implementations are available. It shows similaror better discriminate capability than CART, J48 and REPTree on the typical datasets, reuters-21578, analyzed in this work.

The random forests algorithm provides not only a comparable best classification method, but also some techniques on other statistics, such as out-of-bag estimation, outlier detection, variable importance rank, etc. All these have been implemented in the software for research purpose. Ongoing research includes refinements of the selection method in order to deal appropriately with strongly correlate attributes and the application of RF to other problems in the text classification field.

References

[1] Niusha Shafiabadya, L.H. Leeb, R. Rajkumarc, V.P. Kallimanid, Nik Ahmad Akramc, Dino Isac: Using unsupervised clustering approach to train the Support Vector Machine for text classification. Neurocomputing. Vol.211(2016),p.4-10.

[2] Breiman L:Random Forests.Machine Learning.Vol.45(2001),p.5-32.

[3] Fallon NG , Fielding S, Fernandes PG: Classification of Southern Ocean krill and icefish echoes using random forests. ICES JOURNAL OF MARINE SCIENCE.Vol.72(2016),p.1998-2008.

[4] Breiman L. Bagging Preditors [J]. Machine Learning, 1996, 24(2):123-140.

[5] Luo J, Meng B, Quan CQ, Tu XH: Exploiting salient semantic analysis for information retrieval. ENTERPRISE INFORMATION SYSTEMS.Vol.10(2016),p.959-969.