A Retail Outlet Classification Model Based on AdaBoost

Kai Liu, Bing Wang, Xinshi Lin, Yeyun Ma and Jianqiang Xing

Abstract This paper proposes a framework to get a stable classification rule under unsupervised learning, and the term "stable" means that the rule remains unchanged when the sample set increases. This framework initially makes use of clustering analysis and then use the result of clustering analysis as a reference-studying sample. Secondly, AdaBoost integrated several classification methods is used to classify the samples and get a stable classification rule. To prove the method feasible, this paper shows an empirical study of classifying retail outlets of a tobacco market in a city of China. In this practice, *k*-means is used to make clustering analysis, and AdaBoost integrated RBF neural network, CART, and SVM is used in classification. In the empirical study, this method successfully divides retail outlets into different classes based on the sales ability.

Keywords Retail outlet classification · AdaBoost · Unsupervised learning

1 Introduction

1.1 Background

It is a problem deserving to be discussed for manufacturers that how to decide the amount of goods for each retail outlet on consignment sales or in the planned economy pattern. If the amount of supply is larger than that of demand, the cost for retail outlets to store goods will rise. On the other hand, the period for manufacturers to take back their money will extend. Therefore, an appropriate amount of supply can make work between production and sales more efficient. An approach which can help to decide the amount is predicting present and future sale

K. Liu (✉) · B. Wang · X. Lin · Y. Ma · J. Xing
Xiamen University, Siming S. Rd. 422, Xiamen, China
e-mail: liu.kai@stu.xmu.edu.cn

S. Patnaik and X. Li (eds.), *Proceedings of International Conference on Soft Computing Techniques and Engineering Application*, Advances in Intelligent Systems and Computing 250, DOI: 10.1007/978-81-322-1695-7_42,

condition according to the past sale condition in each retail outlet. However, it is difficult to do such thing for millions of retail outlets.

This paper puts forward a solution. The first step is classifying retail outlets into different kinds according to sales ability of each. Subsequently, specialists work out the amount of goods for each kind of retail outlets through the market analysis. After that, retail outlets that belong to the same kind can receive the same amount of goods. Classification is based on commercial scale and historical sale, which reflect sailing condition. Once the amount has been decided, product launching can be put into practice.

Consequently, this paper emphasizes the problem that how to classify retail outlets and work out an approach to get a classification rule which is stable while the size of samples changes. That way, once there is a new sample, it can be classified into a certain kind according to its features. Since the classification rule remains the same, it can provide a beneficial way for specialists to decide the amount for each kind through market analysis.

1.2 Previous Study

Harald Hruschkaa did a research on market segmentation and compared the performance of self-organizing neural network with *k*-means clustering in 1999 [1]. And the research comes to a conclusion that self-organizing neural network makes applicability more extensive. However, both of these classification rules are changing, while the number of samples is getting larger.

Jih-Jeng Huanga does a research on market segmentation with support vector machine in 2007 [2]. The article gets a classification rule which will not change with the number of samples becoming increasingly large. But the disadvantage is that the classification rule only takes a single classifier, which may not suitable for certain cases.

Data mining algorithms such as decision tree, SVM, *k*-means, neural network have been widely applied to classification and clustering problems, and previous studies [3, 4] have shown that these algorithms are very efficient. In particular, some hybrid models such as bagging and AdaBoost can improve the classification accuracy while comparing to individual model [5, 6]. Therefore, this paper focuses on these algorithms on retail outlets' classification problem.

2 Research Methods

2.1 Overview of the Framework

Because the relative retail outlets which have never been defined into a certain kind before, the classifier has no reference sample to learn from. Clustering can divide samples into different kinds according to some similar features without the

reference-studying sample. Thus, this paper will perform clustering first to get some reference. After clustering, different kinds show the distinction in the ability to sell and the classification result obtained through clustering can be the standard for classifiers as their reference.

Before clustering, what to do beforehand is picking up attributes related to sales ability (otherwise, it can be the wrong classification features with elements which are not related to sales ability). Also, the dimension of sample properties should be reduced to make sure that the number of the attributes of samples is not too much (classifying high-dimensional data lessens the accuracy of the result). At the same time, it assumes that the change in sales ability with the change in the value of selected attributes is monotonous (for example, monotonic increase). It can be expressed in mathematical language. If there is a sample X with some attributes $\{x_1, x_2, \ldots, x_n\}$, function $G(x)$ represents sales ability (though it is not sure that function $G(x)$ can be found in reality), and it is monotonically increasing; thus, it has the following quality (1):

$$\frac{\partial G}{\partial x_i} > 0 \qquad \forall i \in \{1, 2, \ldots, n\} \tag{1}$$

On the other hand, to eliminate dimensional differences, it is necessary to standardize the data, that is,

$$x_i \in [0, 1] \qquad \forall i \in \{1, 2, \ldots, n\} \tag{2}$$

Therefore, sample X can be a point in the space $[0, 1]^n$.

After all these above things have been done, cluster analysis can be carried out with the rectified data. Hopefully, some categories can be worked out, that is, $A = \{C_1, C_2, \ldots, C_k\}$. Each category means different sales ability (or the expected amount of goods to be put into the market). If new samples are added into or old samples are replaced, the result remains the same. To get a classification rule that does not change while the number of samples increased. An approach, which takes clustering first then makes classification based on the clustering result, is considerable. This paper mainly uses AdaBoost algorithm that integrated various kinds of weak classifiers to do ensemble learning so that to get a strong classifier. It takes advantages of different kinds of classifications and makes sure the result to a maximum level of accuracy.

2.2 Cluster Analysis

The main idea is to divide the retail outlets into k sorts to make it convenient for experts to calculate the volume of supply of each retail outlet. Therefore, the cluster method that is suited for this research should generate exactly k clusters as we want. Also, since each retail outlet belongs to a sort, the cluster method should not leave outliers. Thirdly, the distribution of sample set of each cluster should

have a significant difference in space (rather than one encircle another). Thus, the clustering can reflect the difference in sales abilities, which means the supply to the retail outlets we should assign.

Based on the three considerations mentioned above, our research chose *k*-means method to conduct the cluster analysis.

2.2.1 *K*-means Algorithm

K-means clustering is a typical clustering algorithm based on distance. It takes distance as evaluation of similarity.

Given a problem that partition n samples into k clusters C_1, C_2,…, C_k, and the mean of points in each cluster is u_1, u_2, …, u_k, dis(x, y) is the distance between sample x and sample y under a certain measure.

K-means solves this problem by optimizing the following:

$$\text{minimize} \sum_{i=1}^{k} \sum_{x_j \in C_i} \text{dis}(x_j, u_i) \tag{3}$$

This algorithm uses the idea of greedy algorithm. It repeatedly updates the location of mean points in each cluster. Finally, $\sum_{i=1}^{k} \sum_{x_j \in C_i} \text{dis}(x_j, u_i)$ will converge.

The performance of this algorithm depends on the initial mean of points. We can get a better performance by using genetic algorithm or simulated annealing to determine the initial mean of points.

2.3 AdaBoost

AdaBoost is a machine learning algorithm invented by Freund and Schapire [7]. It can composite other machine learning algorithms and improve their performance. AdaBoost algorithm trains the basic classifier (weak classifiers) using multiple training sets and add the result up by their performance to get a stronger final classifier (strong classifier). The theory proved that if each weak classifier performs better than random guesses, the error rate of the strong classifier will converse to zero as the number of weak classifiers goes to infinity [7].

The algorithm generates a different training set by adjusting distribution of weights of each sample, which indicates the importance of each sample in the set for the classification. Let $X = \{x_1, x_2, \ldots, x_n\}$ be the set for the classification and $Y = \{y_1, y_2, \ldots, y_n\}$ be the corresponding class of each sample. $Y = \{y_1, y_2, \ldots, y_n\}$ is a repeated permutation of 1, 2, …, k if samples are divided into k classes. In this paper, we use a vector to represent the distribution of weights (in training round i), say $D_i = \{d_{i,1}, d_{i,2}, \ldots, d_{i,n}\}$, satisfying $\sum_{j=1}^{n} d_{i,j} = 1$.

Suppose we want m samples in training round i, then the training set is S_i, and let the probability be $P(x_j \in S_i) = d_{i,j} \times m$. Therefore, the sample size n_i of training set S_i has the expected value $E(n_i) = m \times \sum d_{i,j} = m$.

At first, each sample has the same weight $d_{0,j} = 1/n, \quad j = 1, 2, \ldots, n$. We generate a training set through this distribution of weights and train a weak classifier named H_1 under this training set.

On this round, the weights of incorrectly classified samples under H_1 are increased, and at the same time, the weights of correctly classified samples are decreased. By doing so, the algorithm gets a new distribution for the weak classifier H_2 in the next round. Thus, H_2 will focus on the samples that H_1 cannot classify. H_1 also gets a weight α_1 based on the error rate it makes, so does H_2. The lower the error rate, the higher the weight. After T training rounds, AdaBoost algorithm produces T weak classifiers and T weights. Finally, the strong classifier is produced by adding T weak classifier up by the corresponding weights.

The algorithm is shown with pseudo-code (Algorithm 1).

initialize $d_{0,j}= 1/n$, $j= 1, 2, \ldots, n$
for $i = \{1, 2, \ldots, T\}$ **do**
  Generate S_i from distribution D_i
  Use the training set S_i to train weak classifier H_i
  Use H_i to do the classify work with sample set X, and calculate the error rate $\epsilon_i = \sum_{H_i(x_j) \neq y_j} d_{i,j}$
  if $\epsilon_i > 1/2$ **then**
    $i = i - 1$
    continue
  end
  Give H_i weight $\alpha_i = \frac{1}{2} \log \frac{1-\epsilon_i}{\epsilon_i}$
  Determine the distribution weight for next training round
  $d_{i+1,j} = \frac{d_{i,j}}{Z_i} \begin{cases} e^{-\alpha_i} & H_i(x_j) = y_j \\ e^{\alpha_i} & H_i(x_j) \neq y_j \end{cases}$, Z_i is the normalizing factor makes $\sum d_{i+1,j} = 1$
end
The strong classifier H is produced by add $H_{1,2,\cdots T}$ up by $\alpha_{1,2,\cdots T}$, $H_{final}(x_j) = a$ if $\sum_i I(H_i(x_j) = a)\alpha_i = max\{\sum_i I(H_i(x_j) = a)\alpha_i | a = 1, 2 \cdots k\}$

Algorithm 1: AdaBoost

2.4 Several Classification Methods

In our research, we take three classification methods, support vector machines (SVM), decision trees, and neural networks (NN), as weak classifier to complete AdaBoost's learning, respectively. These methods have the function to do linear classification, decision tree classifier, and nonlinear classification.

Linear SVM is used to classify the linear classifiable part of our data. As linear SVM can only divide the data into two parts, it needs to do some change while using linear SVM to do the classification work. We use stepwise classification method, solving this problem by classifying 1, 2, …, $k - 1$ classes and k class, and then 1, 2, …, $k - 2$ classes and $k - 1$ class, and so on.

2.4.1 Linear SVM

Linear support vector machine (linear SVM) is a supervised learning model, which classifies the linear classifiable part of our data [8].

The linear SVM uses two hyperplanes $w \cdot x - b = 1$ and $w \cdot x - b = -1$ to separate samples into two parts. And for any x_i, satisfying $(w \cdot x_i - b \leq -1) \vee (w \cdot x_i - b \geq 1)$.

The linear SVM algorithm aims at maximizing the distance between two hyperplanes. By using geometry, we have distance $= 2/|w|$. Thus, the algorithm essentially aims at minimizing $|w|$ [9].

2.4.2 Classification and Regression Tree

The classification and regression tree (CART) is used for our decision tree classification. Each inside node conducts a test on an attribute and has two branches. Each leaf node indicates a class. The samples go through the nodes and fall into exactly one leaf node. The nodes describe the rules which are used to classify the samples [10].

2.4.3 RBF Neural Networks

Radical basis function (RBF) neural networks can solve the nonlinear classification part of our research. RBF neural network is a feedforward neural network that contains the input layer, hidden layer, and output layer. A RBF network is determined by center of radial functions, variance of radial functions, and weights [11].

The training of RBF network maintains these parameters to minimize the error of output layer.

3 Empirical Study

In this paper, we apply the above-mentioned technique to proceed with a precision marketing classification study on a tobacco market of city A, which is located in Fujian Province, China. In the following text, we name it as market A.

3.1 Background

Known as the most important market in Fujian Province, marketing methods of market A include direct sale and consignment sale. However, no matter which method for a company to carry out, the more precise the prediction of the expected tobacco supply volume is, the more efficient the business operations will be.

In the following text, we classify tobacco retail outlets in city A by classifying their sales volume in order to analyze and predict the expected tobacco supply volume of each retail outlet.

The research is based on a database contains most tobacco retail outlets of A, which includes more than 10 thousands samples and upwards of 50 attributes.[1]

3.2 Data Processing

We select five attributes which are mostly related to sales ability among 67 attributes: (1) historical total supply volume, (2) historical total order volume, (3) historical count of making orders, (4) area of counter, and (5) number of employees.

While these attributes are chosen, other attributes such as license number, company code, and name of business corporation are abandoned. And then we normalize these five attributes by mapping (dividing the numbers by their maximum values) vectors from R^5 (which represent attributes of each sample) to $[0, 1]^5$.

It is obvious to find that attribute 1 (historical total supply volume), attribute 2 (historical total order volume), and attribute 3 (historical count of making orders) are of similar index, which reflect the historical sales ability, while area of counter and number of employees are of "hardware" index, which reflect potential sales ability. Here, we use entropy weight method [12, 13] to reduce historical total supply volume, historical total order volume, and historical count of making orders to one index and area of counter and number of employees to another index. And we use these two indexes to proceed with a cluster analysis (however, we still use five attributes to proceed with an AdaBoost classification).

By applying entropy weight method, sample attributes are weighted in accordance with the amount of information of the sample properties and make weight of attributes with a great amount of information. Let sample be x_i, the attribute of sample be x_{ij},

$$p_{ij} = \frac{x_{ij}}{\sum_{i=1}^{n} x_{ij}} \tag{4}$$

entropy will be

[1] Please contact me for the data if anybody wants to continue research on this problem.

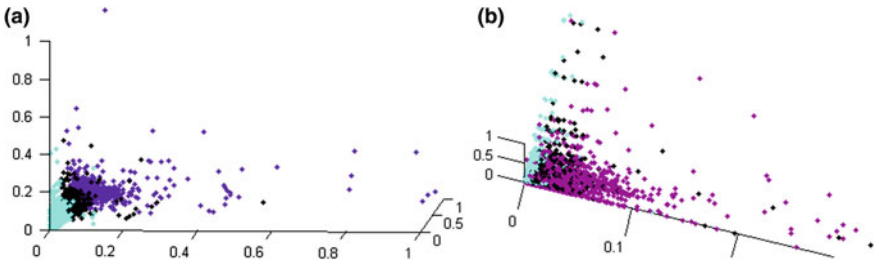

Fig. 1 The result of cluster analysis. **a** The distribution of attributes 1, 3, and 4. **b** The result of cluster analysis 2, 4, and 5

$$e_j = -\frac{1}{\ln(n)} \sum_{i=1}^{n} p_{ij} \ln(p_{ij}) \tag{5}$$

so weight of entropy will be

$$w_j = \frac{1 - e_j}{\sum_k 1 - e_k} \tag{6}$$

Attributes are weighted in accordance with these weights so that we can reduce attributes by mapping vectors from $[0, 1]^5$ (which represent attributes) to $[0, 1]^2$. The economic meanings of these two dimensions are historical sales ability and sales conditions.

3.3 Cluster Analysis

Samples are classified into three categories by using *k*-means algorithm to make a cluster analysis.

Diagrams show distinction of the data. As there are five attributes for the data, so we can only select three attributes to show each time and plot them in the three-dimensional Cartesian coordinate system. We map values of attribute 1 to *x*-axis, attribute 3 to *y*-axis, and attribute 4 to *z*-axis in Fig. 1a and attribute 2 to *x*-axis, attribute 4 to *y*-axis, and attribute 5 to *z*-axis in Fig. 1b. Clusters are separated by different colors.

Samples are well separated into three categories as shown in Fig. 1.

3.4 Classification through AdaBoost

First, we generate several classifiers which are required in AdaBoost learning including classification and regression tree, RBF neural network, and linear SVM

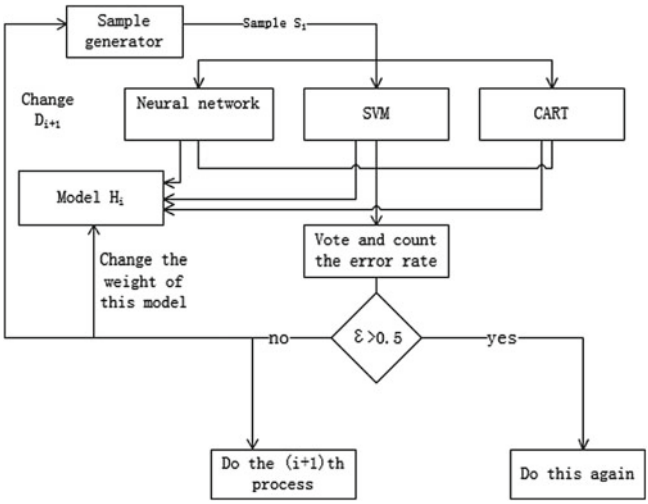

Fig. 2 The *i*th process

(the source code is from the SVM_light, an open source project [14]. Classification and regression tree is compiled as "CART," RBF neural network is compiled as "RBF," and linear SVM is compiled as "SVM." A program that generates the samples according to the distribution of weight is also included and compiled as "sample_generator." A program that calculates the output by voting each sub-output of three classifiers is compiled as "vote."

For loop *i*, "sample_generator" generates a set of data in accordance with D_i. And then, the algorithm calls "SVM," "CART," and "RBF". Each algorithm yields two files as the outputs SVM_data_i.model, SVM_data_i.out, RBF_data_i.model, RBF_data_i.out, CART_data_i.model, CART_data_i.out. The files with suffix ".model" represent classifiers and with ".output" represent the performance of each. Program "vote" takes an equal-weighted vote on SVM_data_i.out, RBF_data_i.out, and CART_data_i.out, which will generate a comprehensive result. The next step is checking the error rate of this classifier generated by "vote"; if error rate is less than 0.5, then we accept this loop of learning. Thereafter, "vote" gives feedback to "sample_generator" and sets the weight for classifiers through the *i*th study. Procedures above are shown in Fig. 2.

We take 11,000 samples out from 11,737 samples for AdaBoost ensemble learning and 737 samples as test samples. Each time we take 5,500 (expected) samples for learning. We conduct a total of 10 times learning. The accuracies[2] of each are shown in Table 1.

[2] Here accuracy rate is defined as 1- error rate. As the sum of d_{ij} is less than than 100 %, the accuracy is slightly larger than the actual accuracy

Table 1 Accuracy

1	2	3	4	5	6	7	8	9	10
99.18 %	99.18 %	99.98 %	98.69 %	100 %	98.96 %	99.87 %	99.81 %	99.83 %	99.04 %

Table 2 Weights of the weak classifiers

1	2	3	4	5	6	7	8	9	10
2.7463	2.7463	4.6051	2.5109	4.6051	2.6269	3.6689	3.4790	3.5347	2.6672

We treat zero error rate as 0.001 % in case of zero division error, which will affect the weighted result. The weights of ten weak classifiers obtained by calculation are shown in Table 2.

At last we classify 11,000 samples used for AdaBoost learning, and the accuracy is 100 %. And then, we classify 737 samples not involved in AdaBoost learning and the accuracy is 99.86 %.The results indicate that AdaBoost integrated learning has a good performance on this study.

4 Conclusion

Actually, our research proposes a framework to solve similar problem, that is, the classification problem with no reference-studying samples and expecting to get a classification rule which will not change over the enlargement of data set. This framework works by taking cluster analysis first, then using the result as reference-studying samples for classifier to learn. Adaboost helps a lot to improve the performance. Empirical study indicates that this method gets a very good result. However, there are many aspects that can be improved under the framework of this paper, and the clustering method and the weak classifier can be changed to fit specific cases.

Acknowledgments We are very grateful to our project mentor Prof. Defu Zhang for his great support on algorithms in the data mining field. This work has been partially supported by the National University Student Innovation Program of China and the National Natural Science Foundation of China (Grant No. 61272003).

References

1. Hruschkaa, H., Natter, M.: Comparing performance of feedforward neural nets and k-means for cluster-based market segmentation. Eur. J. Oper. Res. **114**, 346–353 (1999)
2. Huanga, J.J., Tzeng, G.H., Onga, C.S.: Marketing segmentation using support vector clustering. Expert Syst. Appl. **32**, 313–317 (2007)
3. Zhang, D.F., Chen, Q.S., Wei, L.J.: Building behavior scoring model using genetic algorithm and support vector machines. In: Lecture Notes in Computer Science, vol. 4488, pp. 482–485 (2007)

4. Zhang, D., Leung, S.C.H., Ye, Z.: A decision tree scoring model based on genetic algorithm and k-means algorithm. In: Third International Conference on Convergence and Hybrid Information Technology, vol. 1, pp. 1043–1047 (2008)
5. Zhang, D., Zhou, X., Leung, S.C.H., Zheng, J.: Vertical bagging decision trees model for credit scoring. Expert Syst. Appl. **37**(12), 7838–7843 (2010)
6. Zhang, D., Huang, H., Chen, Q., Jiang, Y.: A comparison study of credit scoring models. In: Third International Conference on Natural Computation, vol. 1, pp. 15–18 (2007)
7. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: CiteSeerX: 10.1.1.56.9855 (1995)
8. Tan, P., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison Wesley Press, Reading (2006)
9. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learning **20**, 273–297 (1995)
10. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth and Brooks/Cole Advanced Books and Software, Monterey (1984)
11. Broomhead, D.S., Lowe, D.: Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical report, p. 4148 (1988)
12. Aczel, J., Daroczy, Z.: On Measures of Information and their Characterizations. Academic Press, New York (1975)
13. Lin, J.: Divergence measures based on the Shannon entropy. IEEE Trans. Inf. Theor. **37**(1), 145–151 (1991)
14. Joachims, T: SVM light-support vector machine, http://svmlight.joachims.org/ (2012). Accessed 10 Aug 2012