Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 436368, 7 pages
http://dx.doi.org/10.1155/2013/436368

Research Article

Classification Based on both Attribute Value Weight and Tuple Weight under the Cloud Computing

Yifeng Zheng, Zaixiang Huang, and Tianzhong He

Department of Computer Science and Engineering, Minnan Normal University, Zhangzhou 363000, China

Correspondence should be addressed to Yifeng Zheng; zhengyifengja@163.com

Received 17 July 2013; Accepted 3 September 2013

Academic Editor: Yuxin Mao

In recent years, more and more people pay attention to cloud computing. Users need to deal with magnanimity data in the cloud computing environment. Classification can predict the need of users from large data in the cloud computing environment. Some traditional classification methods frequently adopt the following two ways. One way is to remove instance after it is covered by a rule, another way is to decrease tuple weight of instance after it is covered by a rule. The quality of these traditional classifiers may be not high. As a result, they cannot achieve high classification accuracy in some data. In this paper, we present a new classification approach, called classification based on both attribute value weight and tuple weight (CATW). CATW is distinguished from some traditional classifiers in two aspects. First, CATW uses both attribute value weight and tuple weight. Second, CATW proposes a new measure to select best attribute values and generate high quality classification rule set. Our experimental results indicate that CATW can achieve higher classification accuracy than some traditional classifiers.

1. Introduction

Cloud computing has become a hot issue in recent years. With the rapid development of information technology and the popularity of cloud computing, it is necessary to mine useful information from magnanimity data [1–9]. Classification is one of the most important tasks in the data mining and the machine learning. Classification can predict the need of users from large data. First, it builds classification rules from training dataset. Second, it uses these rules to predict the class label of new instances.

The traditional classifiers [10–19] frequently adopt the following two ways. Some traditional classifiers remove instance after it is covered by a rule, such as FOIL [20] and ELEM2 [21]. Other traditional classifiers decrease tuple weight of instance after it is covered by a rule, such as PRM and CPAR [22]. Then, we introduce the feature of these classifiers. In the process of extracting rules, FOIL uses measure gain to select a best attribute value and generates one classification rule. It removes instance after it is covered by a rule. As a result, this method is ineffective. It generates a small rule set and cannot achieve high accuracy in some data. ELEM2 uses another measure to generate classification rules. It also removes instance after it is covered by a rule. ELEM2 considers the degree of relevance of an attribute-value pair and selects the most relevant pairs to generate rules. PRM modifies FOIL to achieve higher accuracy. PRM does not remove instance when it is covered by a rule. PRM gives the instance a tuple weight. Thus, PRM can insure that each instance is covered more than once. PRM selects only the best gain to generate rule. CPAR stands in the middle between exhaustive and greedy algorithms and combines the advantages of both. CPAR selects several best attribute values and builds several rules at one time. It does not remove instance immediately when it is covered by a rule. CPAR also uses tuple weight to guarantee that each instance can be covered more than once. These methods do not employ attribute value weight. They cannot get high quality classification rule set. As a result, they can not achieve high classification accuracy in some data.

In this paper, we propose a new algorithm, named classification based on both attribute value weight and tuple weight (CATW). CATW uses the both attribute value weight and tuple weight. Moreover, CATW uses a new measure to improve the quality of classification rule set. Our method has following advantages.

(1) After an instance is covered by a rule, instead of removing it, its weight is decreased by multiplying a factor. Thus, we can guarantee that each instance can be covered more than once.

(2) If we only use tuple weight, we cannot change the importance of an attribute-value pair in the dataset. Therefore, CATW uses attribute value weight to reduce the importance of attribute-value pair after the rule is generated. In this way, CATW can increase the chances of attaining other optimal attribute-value pairs. We can generate more high quality of rules.

(3) CATW presents a new measure to select the best attribute value. CATW uses two different measures: support and correlation confidence. If two different attribute-value pairs have same correlation confidence, CATW considers their support.

Experimental results indicate that: (1) if the instance is removed immediately after it is covered by a rule, the classifier generates a very small number of rules; (2) if the classifier is only using tuple weight, the quality of classification rule set is not good. (3) Since CATW uses both attribute value weight and tuple weight, it achieves high classification accuracy.

The outline of this paper is as follows. Section 2 presents the details of CATW and describes the process of rule generation in CATW. Section 3 discusses how to predict class label using the rules. The experimental results are presented in Section 4. Finally, we conclude the study in Section 5.

2. Rule Generation of CATW

The algorithm of CATW has three special points: the attribute value weight, the tuple weight, and the improved measure. First, we describe the method of how to use tuple weight. Second, we introduce the use of attribute weight. Third, we propose a new measure to generate high quality classification rule set. Finally, we show the whole process of how to generate rule set.

Let T be a set of tuples. Each tuple t has attributes $\{A_1, A_2, \ldots, A_m\}$. Suppose C to be a set of class labels $\{C_1, C_2, \ldots, C_k\}$, where k means the number of class label.

Definition 1 (a literal). A literal p is an attribute-value pair, which follows the pattern of (A_i, v), where A_i is an attribute and v is a value of attribute A_i.

Definition 2 (a classification rule). $X \rightarrow c$ is called a classification rule r, if X consists of a conjunction of literals $p_1, p_2, \ldots, p_l$ with the form of $p_1 \land p_2 \land \cdots \land p_l$, where c is a class label.

A tuple t satisfies the antecedent of r if and only if it has all literals in r. If t satisfies the antecedent of r, r predicts that t has a class label c.

2.1. The Tuple Weight. In traditional classification, all rules are generated from the training database. If a tuple t is covered by a rule r, they can not ensure that r is the best rule for t. If r is generated from the remaining dataset instead of the whole dataset [22], r may not be the best rule. In order to improve the classification accuracy and increase the number of rules, some traditional classifiers use tuple weight. By depending on tuple weight, these classifiers can delay removing instance after it is covered by a rule. In our algorithm, after a tuple is covered by a rule, instead of removing it, its weight is decreased by multiplying a factor. We set a threshold for tuple weight. When the tuple weight of tuple t is less than threshold, we remove the tuple t from training data. CATW produces more rules. Each tuple can be covered by classification rules more than once.

In our approach, we can set an initial threshold and an end threshold. We can limit the number of rules which are generated according to actual situation. If we set a small end threshold, it generates a large number of rules. On the contrary, if we set a large end threshold, it generates a less number of rules. In our experiment, we set an initial threshold 1, a weight factor 0.75. Moreover, we set an end threshold. The end threshold is the third power of weight factor. We can make sure that each instance can be covered three times.

2.2. The Attribute Value Weight. Some traditional classifiers only use tuple weight. They do not change the importance of an attribute-value pair in the training data. After a rule is generated, these classifiers may select the duplicate attribute-value pair. Thus, they may miss some high quality rules which can be used to affect the classification accuracy. CATW uses attribute value weight to reduce the importance of attribute-value pair after the rule is generated. When the tuple is covered by a rule, our algorithm can reduce the importance of attribute-value pairs which are contained in it. In this way, we can increase the chances of attaining another optimal attribute-value pair.

Example 3. The following training dataset with two classes is shown in Table 1. Then, we demonstrate how to use attribute value weight.

Suppose r = {OUTLOOK = rain $\land$ WINDY = TRUE $\rightarrow$ PLAY = no} to be just generated. Then, we set a weight factor 0.8, and set {PLAY = no} for positive examples. After a rule is generated, CATW uses weight factor to reduce the importance of all attribute values that are contained in antecedent of the rule in positive examples. The result is shown in Table 2.

The results of our experiment indicate that classification accuracy is influenced by attribute value weight. Compared with the classifiers which do not use attribute value weight, CATW can achieve higher classification accuracy in some data. Thus, the attribute value weight can be a help to improve the quality of classification rule.

2.3. The Measure of CATW. Some classifiers use FOIL gain to select literal. FOIL gain is used to measure the information gained from adding literal p to the current rule. Let us suppose that $|P|$ means the number of positive examples which satisfies the antecedent of the current rule r and $|N|$

Table 1: The training dataset.

	Outlook	Temperature	Humidity	Windy	Play
	Sunny	Hot	>75	False	No
AW	1	1	1	1	
	Sunny	Hot	>75	True	No
AW	1	1	1	1	
	Overcast	Hot	>75	False	Yes
AW	1	1	1	1	
	Rain	Mild	>75	False	Yes
AW	1	1	1	1	
	Rain	Cool	>75	False	Yes
AW	1	1	1	1	
	Rain	Cool	≤75	True	No
AW	1	1	1	1	
	Sunny	Mild	≤75	True	Yes
AW	1	1	1	1	

Table 2: Attribute value weight in positive examples.

	Outlook	Temperature	Humidity	Windy	Play
	Sunny	Hot	>75	False	No
AW	1	1	1	1	
	Sunny	Hot	>75	True	No
AW	1	1	1	0.8	
	Overcast	Hot	>75	False	Yes
AW	1	1	1	1	
	Rain	Mild	>75	False	Yes
AW	1	1	1	1	
	Rain	Cool	>75	False	Yes
AW	1	1	1	1	
	Rain	Cool	≤75	True	No
AW	0.8	1	1	0.8	
	Sunny	Mild	≤75	True	Yes
AW	1	1	1	1	

Table 3: Characteristics of UCI datasets.

Dataset	No. of instances	No. of attributes	No. of class
Auto	205	25	7
Cleve	303	13	2
Glass	214	9	7
Heart	270	13	2
Hepati	155	19	2
Horse	368	22	2
Iono	351	34	2
Iris	150	4	3
Labor	57	16	2
Lymph	148	18	4
Wine	178	13	3
Zoo	101	16	7

means the number of negative examples which satisfy the antecedent of the current rule r. After literal p is added to r, $|P^*|$ means the number of positive examples which satisfy the antecedent of the new rule, and $|N^*|$ means the number of negative examples which satisfy the antecedent of the new rule [22]. The FOIL gain of p is defined as:

$$\text{gain}(p) = |P^*| \left(\log\left(\frac{|P^*|}{(|P^*| + |N^*|)} \right) - \log\left(\frac{|P|}{(|P| + |N|)} \right) \right). \quad (1)$$

In our experiment, we employ two different improved measures.

2.3.1. Improved FOIL Measure. In our experiment, $|P|$ means total tuple weight of positive examples which satisfy the antecedent of current rule r. $|N|$ means total tuple weight of negative examples which satisfy the antecedent of current rule r. After literal p is added to r, $|P^*|$ means total attribute value weight of literal p in positive examples, and $|N^*|$ means total attribute value weight of literal p in negative examples. Therefore, CATW uses both tuple weight and attribute value weight when it measures literal p. We call this measure an improved FOIL measure.

2.3.2. Improved Correlation Measure. In traditional FOIL gain, $|P^*|$ has a huge influence to select a best attribute value. For example, if $\log(|P^*|/(|P^*| + |N^*|)) - \log(|P|/(|P| + |N|))$ is too small and $|P^*|$ is too large, the result of gian(p) is not the best for rule. We use two different measures: support and correlation confidence. We divide the traditional FOIL measure in two parts.

(1) PART I: $|P^*|$.

(2) PART II: $\log(|P^*|/(|P^*|+|N^*|))-\log(|P|/(|P|+|N|))$.

When we select literal p, a global order of literal p is composed. Given two literal p_1 and p_2, p_1 is better than p_2, denoted as $p_1 > p_2$.

$p_1 > p_2$ if and only if (1) PART II (p_1) > PART II (p_2) or (2) PART II (p_1) = PART II (p_2) and PART I (p_1) > PART I (p_2). We call this measure an improved correlation measure.

2.4. Algorithm of CATW. In this part, we will introduce our algorithm in detail. The CATW algorithm is presented in Algorithm 1.

3. Classification of CATW

Before making any prediction, we use the Laplace expected error estimate [23] to evaluate the quality of rules. It is defined as follows:

$$\text{Laplace accuracy} = \frac{(n_c + 1)}{(n_{\text{tot}} + k)}, \quad (2)$$

where k is the number of classes and n_{tot} is the total number of examples satisfying the antecedent of rule, among which n_c examples belong to c.

Table 4: The accuracy of CATW with improved FOIL gain measure.

	FOIL	CMAR	CPAR	TW	AW(0.8)/TW	AW(0.5)/TW
Auto	0.776	0.781	0.82	0.7984	0.7934	0.7883
Cleve	0.7423	0.822	0.815	0.7695	0.7907	0.8014
Glass	0.7156	0.701	0.744	0.7385	0.7481	0.7481
Heart	0.8148	0.822	0.826	0.8214	0.8095	0.8095
Hepati	0.78	0.805	0.794	0.8444	0.8579	0.8705
Horse	0.7124	0.826	0.842	0.7856	0.7915	0.8032
Iono	0.889	0.915	0.926	0.9109	0.9293	0.9263
Iris	0.9533	0.94	0.947	0.9583	0.9583	0.9583
Labor	0.7567	0.897	0.847	0.8148	0.9206	0.9365
Lymph	0.7424	0.831	0.823	0.8157	0.8380	0.8454
Wine	0.9379	0.95	0.955	0.9526	0.9708	0.9708
Zoo	0.9409	0.971	0.951	0.9503	0.9310	0.9310
Average	0.8134	0.8551	0.8575	0.8467	0.8616	0.8658

Table 5: The accuracy of CATW with improved correlation measure.

	FOIL	CMAR	CPAR	TW	AW(0.8)/TW	AW(0.5)/TW
Auto	0.776	0.781	0.82	0.7927	0.8054	0.7773
Cleve	0.7423	0.822	0.815	0.7941	0.7945	0.8227
Glass	0.7156	0.701	0.744	0.7385	0.7385	0.7433
Heart	0.8148	0.822	0.826	0.8056	0.8333	0.8294
Hepati	0.78	0.805	0.794	0.8305	0.8370	0.8644
Horse	0.7124	0.826	0.842	0.7236	0.7942	0.7915
Iono	0.889	0.915	0.926	0.9137	0.9352	0.9322
Iris	0.9533	0.94	0.947	0.9583	0.9583	0.9583
Labor	0.7567	0.897	0.847	0.918	0.9180	0.9365
Lymph	0.7424	0.831	0.823	0.7676	0.8310	0.8380
Wine	0.9379	0.95	0.955	0.9646	0.9646	0.9532
Zoo	0.9409	0.971	0.951	0.9495	0.9697	0.9596
Average	0.8134	0.8551	0.8575	0.8464	0.865	0.8672

When using rules to predict the class-label of unknown instance, we use several rules which are matched by the instance. If all the rules have the same consequent of rule, we assign that label to the instance. If all the best rules have several classes, we calculate the average Laplace accuracy of each class. Then, we select the class label with the highest average value and assign it to the instance.

4. Experimental Results

All experiments are performed on 12 different datasets from the UCI data collection. All datasets were conducted using stratified tenfold cross-validation. In cross-validation, the data set is divided into 10 blocks. Each block is held out once. The classifier is trained on the remaining 9 blocks. The character of each dataset is shown in Table 3. We perform our experiments on a 2.2 GHz PC with 2 G memory, running Microsoft Windows XP.

In Tables 4 and 5, Column 1 shows the accuracy of FOIL. Column 2 shows the accuracy of CMAR. Column 3 shows the accuracy of CPAR. Column 4 shows the accuracy of CATW without attribute value weight, set tuple weight 0.75. Column 5 shows the accuracy of CATW, set attribute value weight 0.8 and tuple weight 0.75. Column 6 shows the accuracy of CATW, set attribute value weight 0.5 and tuple weight 0.75.

In Table 4, we use the measure which is an improved FOIL measure. Figure 1 gives the accuracy of FOIL, CMAR, CPAR, and CATW based on Table 4. CATW uses both attribute value weight and tuple weight and employs the improved FOIL measure. From Figure 1 and Table 4, we can see that CATW can achieve higher accuracy than FOIL, CMAR, and CPAR.

In Table 5, we use the measure which is an improved correlation measure. Figure 2 gives the accuracy of FOIL, CMAR, CPAR, and CATW based on Table 5. CATW uses both attribute value weight and tuple weight and employs the improved correlation measure. From Figure 2 and Table 5, we can see that CATW with improved correlation measure can also achieve higher accuracy than FOIL, CMAR, and CPAR.

By comparison, the accuracy of CATW with the improved correlation measure is higher than the accuracy of CATW with the improved FOIL measure. From Tables 4 and 5, we can see that it is necessary to use the improved correlation measure.

Table 6: Comparing different attribute value weights with improved FOIL gain measure.

	AW(0.8)/TW	AW(0.75)/TW	AW(0.67)/TW	AW(0.5)/TW	AW(0.33)/TW
Auto	0.7934	0.7881	0.7730	0.7883	0.8035
Cleve	0.7907	0.7946	0.7874	0.8014	0.7836
Glass	0.7481	0.7481	0.7530	0.7481	0.7433
Heart	0.8095	0.8135	0.8056	0.8095	0.8054
Hepati	0.8579	0.8640	0.8709	0.8705	0.8239
Horse	0.7915	0.7735	0.7884	0.8032	0.7971
Iono	0.9293	0.9230	0.9294	0.9263	0.9263
Iris	0.9583	0.9583	0.9583	0.9583	0.9583
Labor	0.9206	0.9206	0.9048	0.9365	0.9048
Lymph	0.8380	0.8181	0.8449	0.8454	0.8523
Wine	0.9708	0.9766	0.9708	0.9708	0.9766
Zoo	0.9310	0.9310	0.9310	0.9310	0.9209
Average	0.8616	0.8591	0.8598	0.8658	0.858

Table 7: Comparing different attribute value weights with improved correlation measure.

	AW(0.8)/TW	AW(0.75)/TW	AW(0.67)/TW	AW(0.5)/TW	AW(0.33)/TW
Auto	0.8054	0.7876	0.7720	0.7773	0.7670
Cleve	0.7945	0.8085	0.8087	0.8227	0.8264
Glass	0.7385	0.7283	0.7431	0.7433	0.7431
Heart	0.8333	0.8254	0.8294	0.8294	0.8056
Hepati	0.8370	0.8574	0.8439	0.8644	0.8513
Horse	0.7942	0.7913	0.7915	0.7915	0.7738
Iono	0.9352	0.9261	0.9322	0.9322	0.9353
Iris	0.9583	0.9583	0.9583	0.9583	0.9583
Labor	0.9180	0.9180	0.9339	0.9365	0.9180
Lymph	0.8310	0.8028	0.8241	0.8380	0.8245
Wine	0.9646	0.9529	0.9412	0.9532	0.9587
Zoo	0.9697	0.9596	0.9596	0.9596	0.9495
Average	0.865	0.8597	0.8615	0.8672	0.8593

Table 6 displays the accuracy of different attribute value weights in CATW. In Table 6, CATW employs the improved FOIL measure. Table 7 displays the accuracy of different attribute value weights in CATW. In Table 7, CATW employs the improved correlation measure. The results of the two tables indicate that (1) the accuracy of improved correlation measure is higher than the accuracy of improved FOIL measure and (2) different value of attribute value weight has different influence on the accuracy of classification.

Through all the above results of our experiment, we can conclude that (1) it is necessary to use attribute value weight and tuple weight; (2) it is necessary to use improved correlation measure; (3) different value of attribute value weight has different influence on the accuracy of classification.

5. Conclusions and Future Work

With the rapid development of information technology and the popularity of cloud computing, it is necessary to mine useful information from magnanimity data. Some traditional classification methods frequently adopt the following two ways. One way is that it does not use tuple weight to remove instance after it is covered by a rule. Another way is that it only gives tuple weight of instance after it is covered by a rule. As result, they cannot achieve high classification accuracy in some data. In this paper, we present a novel approach CATW. First, CATW uses both attribute value weight and tuple weight. Second, CATW proposes a new measure which is the improved correlation measure. CATW employs the improved correlation measure to select best attribute values and generate high quality classification rule set. The results of our experiment indicate that CATW can generate a reasonable number of classification rules. In addition, CATW can achieve high classification accuracy. Our experiment shows that different value of attribute value weight has different influence on the accuracy of classification. At present, we cannot find the regular change in selecting an optimal attribute value weight. In future research, we will focus on it. We also focus on another research. We will combine distributed data mining with cloud computing platform in order to improve the efficiency of CATW.

```
Input: Training set $D = P \cup N$ ($P$ and $N$ are the sets of all positive and negative example, respectively)
Output: A set of rules for predicting class labels for examples
Procedure CATW
    attributeWeight $\leftarrow \alpha$
    tupleWeight $\leftarrow \beta$
    tupleThreshold $\leftarrow \lambda$
    rules $\leftarrow$ null
    while $|P| > 0$
        $N' \leftarrow N, P' \leftarrow P$
        $r \leftarrow$ null
        while $|N| > 0$ and r.length < max_rule_length
            find the best attribute value av use the improved correlation measure combine tuple weight with attribute weight
            add av to $r$
            remove from $P'$ all examples not satisfying $r$
            remove from $N'$ all examples not satisfying $r$
        end
        add $r$ to rules
        for each attribute at that is included in antecedent of $r$ in $P$
            at.weight $\leftarrow$ attributeWeight $*$ at.weight
        end
        for each example $t$ in $P$ satisfying $r$'s body
            t.weight $\leftarrow$ tupleWeight $*$ t.weight
            if t.weight < tupleThreshold then remove $t$ from $P$
        end
    end
return rules
```

Algorithm 1: Classification based on both attribute value weight and tuple weight (CATW).

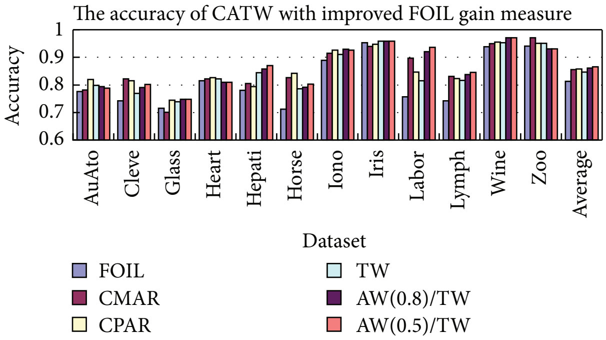

Figure 1: The accuracy of CATW with improved FOIL gain measure.

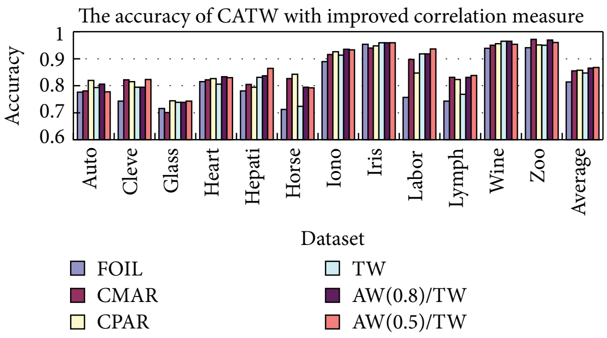

Figure 2: The accuracy of CATW with improved correlation measure.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work is funded by the China NFS Program (no. 61170129), and by the Fujian province NSF Program (no. 2013J01259).

References

[1] K. Lal and N. C. Mahanti, "A novel data mining algorithm for semantic web based data cloud," *International Journal of Computer Science and Security*, vol. 4, no. 2, pp. 160–175, 2010.

[2] S. Adapa, M. Kalyan Srinivas, and A. V. R. K. Harsha Vardhan Varma, "A study on cloud computing data mining," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 1, no. 5, pp. 1232–1237, 2013.

[3] Z. Qureshi, J. Bansal, and S. Bansal, "A survey on association rule mining in cloud computing," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 4, pp. 318–321, 2013.

[4] J. Ding and S. Yang, "Classification rules mining model with genetic algorithm in cloud computing," *International Journal of Computer Applications*, vol. 48, no. 18, pp. 24–32, 2012.

[5] L. Hu, Z. Zhang, F. Wang, and K. Zhao, "Optimization of the deployment of temperature nodes based on linear programing

in the internet of things," *Tsinghua Science and Technology*, vol. 18, no. 3, pp. 250–258, 2013.

[6] S. Gond, A. Patil, and V. B. Nikam, "A survey on parallelization of data mining techniques," *International Journal of Engineering Research and Applications*, vol. 3, no. 4, pp. 520–526, 2013.

[7] N. Mishra, S. Sharma, and A. Pandey, "High performance cloud data mining algorithm and data mining in clouds," *IOSR Journal of Computer Engineering*, vol. 8, no. 4, pp. 54–61, 2013.

[8] A. Pareek and M. Gupta, "Review of data mining techniques in cloud computing database," *International Journal of Advanced Computer Research*, vol. 2, no. 2, pp. 52–55, 2012.

[9] R.-Ş. Petre, "Data mining in cloud computing," *Database Systems Journal*, vol. 3, no. 3, pp. 67–71, 2012.

[10] Y. Jiao, "Research of an improved apriori algorithm in data mining association rules," in *Proceedings of the IEEE International Conference on Information Theory and Information Security (ICITIS '11)*, November 2011.

[11] F. Thabtah, P. Cowling, and Y. Peng, "MCAR: multi-class classification based on association rule," in *Proceedings of the 3rd ACS/IEEE International Conference on Computer Systems and Applications*, pp. 127–133, January 2005.

[12] G. Dong, X. Zhang, L. Wong, and J. Li, "CAEP: classification by aggregating emerging patterns," *Discovery Science*, vol. 1721, pp. 30–42, 1999.

[13] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *Proceedings of the ACM SIGMOD international Conference on Management of Data (SIGMOD '00)*, pp. 1–12, 2000.

[14] W. Li, J. Han, and J. Pei, "CMAR: accurate and efficient classification based on multiple class-association rules," in *Proceedings of the 1st IEEE International Conference on Data Mining (ICDM '01)*, pp. 369–376, San Jose, Calif, USA, November 2001.

[15] F. A. Thabtah and P. I. Cowling, "A greedy classification algorithm based on association rule," *Applied Soft Computing Journal*, vol. 7, no. 3, pp. 1102–1111, 2007.

[16] B. Liu, W. Hsu, and Y. Ma, "Integerating classification and association rule mining," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD '98)*, pp. 80–86, New York, NY, USA, August 1998.

[17] X. Wang, Z. Zhou, and G. Pan, "CMER: classification based on multiple excellent rules," *Journal of Theoretical and Applied Information Technology*, vol. 48, pp. 661–665, 2013.

[18] G. Chen, H. Liu, L. Yu, Q. Wei, and X. Zhang, "A new approach to classification based on association rule mining," *Journal of Decision Support Systems*, vol. 42, no. 2, pp. 674–689, 2006.

[19] P. Leng and F. Coenen, "The effect of threshold values on association rule based classification accuracy," *Journal of Data and Knowledge Engineering*, vol. 60, no. 2, pp. 345–360, 2007.

[20] J. Ross Quinlan and R. Mike Cameron-Jones, "FOIL: a midtern report," in *Proceedings of the European Conference Machine Learning*, pp. 3–20, Vienna, Austria, 1993.

[21] A. An, "Learning classification rules from data," *Computers & Mathematics with Applications*, vol. 45, no. 4-5, pp. 737–748, 2003.

[22] X. Yin and J. Han, "CPAR: classification based on predictive association rules," in *Proceedings of the Society for Industrial and Applied Mathematics (SIAM) International Conference on Data Mining*, May 2003.

[23] P. Clark and R. Boswell, "Rule induction with CN2: somerecent improvements," in *Proceedings of European Working Session on Learning (EWSL '91)*, pp. 151–163, Porto, Portugal, March 1991.