Learning Classification Rules Based on Effect Measure

Tianzhong He, Zhongmei Zhou and Zaixiang Huang

Department of Computer Science and Engineering, Minnan Normal University, Zhangzhou 363000, China.
hetianzhong@163.com

Abstract

To produce a new Rule, many rule-based classifiers use one measure to select the best attribute-value pair. So, a lot of attribute-values have the same best values, and we cannot distinguish which attribute-value pair is the best. On the other hand, these classifiers usually combine the best attribute-value pairs to produce rules, whether they bias toward the same class label or not. To address issues, this paper proposes a new measure approach named deviation. Using the attribute-value deviation, it is easy to distinguish which attribute-value pairs bias toward the same class label. In this paper, we propose a multi-measure called effect measure to select the best attribute-value pair. It integrates the deviation and the entropy, and we also propose a new classification approach called CAEM which uses the effect measure to select the best attribute-value pairs. Experimental results show the method of multi-measure is necessary.

Keywords: *Classification, Entropy, Multi-measure, Deviation, Effect*

1. Introduction

sAs one of the most fundamental data mining tasks, classification has been extensively studied. Recently, many classification approaches have been proposed, such as decision tree [1], rough set approaches [2-4], associative classification [5-7], KNN [8-10] and other approaches [11]. Among them, one category is the rule-based classifiers [1, 12-14]. They build a model from the training database as a set of high-quality rules, which can be used to predict the class labels of unlabeled instances. Many studies have shown that rule-based classification algorithms perform well in classifying categorical databases [13, 7, 5, 15]. However, these algorithms may suffer from two major shortages.

First, many rule-based algorithms [1, 12, 13] discover a set of classification rules using a single measure to select a best attribute-value pair. Attribute-value pairs often have the same best value according a single measure. For example, C4.5 [1] only uses the attribute information gain to select a best attribute-value pair. PRM [12] only uses Foil gain to select a best attribute-value pair. Unfortunately, it is inappropriate to select a best attribute-value pair according to a single measure. In other words, with single measure, the best attribute-value pair can not be selected correctly.

Second, these rule-based algorithms induce rules directly from all attribute-value pair space. FOIL[] and PRM repeatedly searches for the current best rule and removes all the positive instances covered by the rule until all the positive instances in the data set are covered. The main drawback of this approach is that the attribute-value pairs of a rule do not bias toward the same class label.

In this paper, we proposed a new approach, called Classification based on Attribute-value pair Effect Measure (CAEM), to address these issues. First, CAEM groups all attribute-value pairs by their deviations. So, attribute-values in one group bias to the same class label. Second,

CAEM uses a new multi measure method to select the best attribute-value pair, which integrates attribute-values pair entropy and attribute-value pair deviations to select the best attribute-value pair. In comparison with single measure, multi measures can greatly decrease the number of attribute-value pair with same best values. Rules are generated from the groups respectively. Third, when an uncertain class label example satisfies a set of rules, CAEM adopts rule's strength to classify it. Our experimental results show that the techniques developed in this paper is better performance than one measure.

The rest of the paper is organized as follows. Section 2 introduces the effect measure of the attribute-value. Section 3, we describe classification based on Attribute-value pair effect measure. The experimental results are presented in Section 4, and we conclude the study in Section 5.

2. The Effect Measure of Attribute-value

A set of tuples T has m distinct attributes A_1, A_2, A_m and class C = {c_1, c_2, …, c_k}. Each tuple {a_1, a_2, … , a_m, c}, called instance, where a_i is the value of the A_i and c is the value of C . The entropy of an attribute-value pair (A_i, a_i) is defined by

$$entropy(A_i, a_i) = \sum_{j=1}^{k} -\left(\frac{m_j}{n} \log\left(\frac{m_j}{n} \right) \right), \tag{2.1}$$

where n is the number of the instances which satisfy the attribute-value pair (A_i, a_i). m_j is the number of instances which satisfy attribute-value pair (A_i, a_i) and the class is c_j.

The attribute-value pair (A_i, a_i) deviation to the class c_1 is defined by

$$deviation_{c1}(A_i, a_i) = \frac{2}{n}(m_1 - m_2), \tag{2.2}$$

where n is the number of all instances. m_1, m_2 is the number of instances which satisfy attribute-value pair (A_i, a_i) and the class is c_1, c_2.

The formula (2.2) only applies to the binary class. The extend formula for the multi-class is defined by

$$deviation_c(A_i, a_i) = \frac{2}{n}\left(m_c - \frac{m_o}{k-1} \right), \tag{2.3}$$

where n is the number of all instances. m_c is the number of instances which satisfy attribute-value pair (A_i, a_i) and the class is c. m_o is the number of instances which satisfy attribute-value pair (A_i, a_i) and the class is not c. Obviously, the formula (2.3) contains the formula (2.2).

The attribute-value pair (A_i, a_i) effect to the class c is defined by

$$effect_c(A_i, a_i) = \lambda(1 - entropy(A_i, a_i) + (1-\lambda) deviation_c(A_i, a_i), \tag{2.4}$$

From the formula (2.4), we can know the effect measure integrates the deviation and the entropy. So, it not only selects the best attribute-value but also distinguishes which class label the attribute-value pair bias to.

Rule-based algorithms work in three phases:

First, rule-based algorithms employ a function for evaluating attribute-value pairs. Various evaluation criteria have been used in different learning algorithms. For example, C4.5 [1] employs an entropy-based information gain to find the most relevant

attribute to grow decision trees. PRISM [15] uses another form of information gain which can be characterized in terms of apparent classification accuracies on the training set to measure the relevance of attribute-value pairs with respect to a target concept. They all only use single measure to find there relevant of attribute-value pairs which leads to suffer from selecting the best attribute-value pair. We use a multi measure method to select the best attribute-value pair, which integrates attribute-values pair entropy and attribute-value pair deviations.

Second, after evaluating attributes or attribute-value pairs, these rule based algorithms extract rules directly from all attribute-value pair's space. Obviously, it is not appropriate. Instead of searching from all attribute-value pair's space, we divide all attribute-value pairs into groups, and generate rules from the groups respectively.

Third, to predict a new instance's class label, some algorithms use the first matching rule [5] and others use multiple rules [7]. For matching rules, there are different ways to combine these rules to classify the new instance. Some algorithms average the confidences for each category, while others compute a weighted chi-square [7] for each category. We use the rule-strength to measure a rule, which integrates the laplace expected error estimate [13] and support of the rule, and average the rule-strengths for each category. From the experiments, our algorithm achieved high classification accuracy

3. Classification based on the Attribute-value Pair Effect Measure

In this section, we develop a new rule-based classification method, called Classification based on the Attribute-value pair Effect Measure (CAEM). To produce rules, CAEM conducts the following.

1. Group all attribute-value pairs by the deviation.

2. Calculate the effects of all attribute-value pairs for each group.

3. Produce rules for the groups respectively.

The general idea of CAEM is shown in the following example. Consider the data set as shown in Table 1. The data set is the subset of mushrooms. The last attribute is class. In this example, λ is set to 0.5.

3.1. Group all Attribute-value Pairs by their Deviation

For a class c, CAEM calculate all attribute-value pair's deviation, shown in the Table 2. If the deviation is greater than zero, the attribute-value belongs to class c group. Groups of this data set are shown in the Table 3 and Table 4. The formula (2.3) shows that deviation reflects the difference of the instances which belongs to the class c and does not belong to the class c. The larger deviation is, the greater the contribution to the class c. The zero is that attribute-value pairs neither bias to class c not to other class. The negative value means the attribute-value is not bias to class c. So, grouping all attribute-value pairs by their deviation is reasonable.

Table 1. A Training Data Set

instance	A	B	C	D	W
x1	32	55	80	83	90
x2	33	52	80	85	89
x3	33	52	80	85	89
x4	33	55	79	82	90
x5	34	55	79	82	89
x6	34	55	77	82	89
x7	32	55	80	88	90
x8	33	55	79	82	90

Table 2. Deviation for Class 89

Attr	Value	Deviation
A	32	0.5
A	33	0
A	34	-0.5
B	55	0.5
B	52	-0.5
C	80	0
C	79	0.25
C	77	-0.25
D	83	0.25
D	85	-0.5
D	82	0
D	88	0.25

Table 3. The Group ofClass 89

Attr	Value	Deviation
A	32	0.5
B	55	0.5
C	79	0.25
D	83	0.25
D	88	0.25

Table 4. The Group of Class 90

Attr	Value	Deviation
A	34	0.5
B	52	0.5
C	77	0.25
D	85	0.5

3.2. Calculate all Attribute-value Pair's Effects in the Group

After Grouping, CAEM calculate all attribute-value pair's effects, the effect to 89 is shown in the Table 5. As The Table 5 shows, some attribute-value pairs can be not distinguished

with one measure, such as the pairs (A, 32) and (B, 55). But we can distinguish them by effect. Obviously, it is appropriate to select a best attribute-value pair according to a multi-measure.

Table 5. Calculate Effects

Attr	Value	Deviation	Entropy	Effect
A	32	0.5	0	0.75
B	55	0.5	0.92	0.29
C	79	0.25	0.92	0.165
D	83	0.25	0	0.625
D	88	0.25	0	0.625

3.2. Producing Rules

After grouping, CAEM produce rules for the groups respectively. The producing rules algorithm is presented in Figure 1.

In this example, min-conf is set to 0.8, min-effect is set to 0.02, δ is set to 0.1, α is set to 0.2, and β is set to 0.01.

CAEM selects the best attribute-value pair to the rule, then changes it's effect, until the rule' confidence achieves min-conf. For the Table 5, CAEM selects (A, 32), (D, 83) and (D, 88) and their confidence are 1. CAEM produces three rules as follow:

r1: A = 32 =>W = 90, r2: D = 32=> W = 90, r3: D = 88=> W = 90.

After producing three rules, effects are changed as shown Table 6. Current-weight is 2.51. Being greater than 1.6, producing rules continues. After (B, 55) and (C, 79) selected, rule confidence is 0.67, less than min-conf, but the max effect is 0.0075 which less than min-effect. So, producing rule for class 89 group stops. To learn rules for other group, the procedure repeats. The class 89 group after the attribute-value pair (C, 79) selected as shown Table 7.

Table 6. After Producing Three Rules

Attr	Value	Effect
A	32	0.0075
B	55	0.29
C	79	0.165
D	83	0.00625
D	88	0.00625

Table 7. After (C, 79) Selected

Attr	Value	Effect
A	32	0.0075
B	55	0.0029
C	79	0.00165
D	83	0.00625
D	88	0.00625

Input: Dataset D.
Output: A rule set for predicting class labels for instances.
1 Initialize Options;
2 R←Φ;
3 count the deviations for all attribute-vales

```
4 divide all attribute-values into groups by their deviations
5 for each g in groups do
6     for each instance ins in the dataset D,ins.weight      ←1;
7     initWeigth              Weight();
8     while (currentWeight() > α* initWeight) do
9        fromAttr                    ← all attribute;
10        r      ←Φ;
11       while fromAttr ≠Φ do
12          attribute-value av = max(g, fromAttr);
13          if av.effect < minEffect then
14            continue for;
15          end if
16          append av to r;
17       delete the attribute of the av from fromAttr;
18          count the rule r'confidence rConf
19          if (rConf ≥ minConfidence) then
20            for all instances ins satisfying r do
21              ins.weight =δ*ins.weight;
22            end for
23          end if
24         av.effect =β*av.effect;
25          R          ←R∪r;
26       end while
27    end while
28 end for
29 return R;
```

Figure 1. CAEM Algorithm

3.3 Classification using Rules

CAEM collects the subset of rules matching the new instance from the set of rules. If all the rules matching the new instance have the same class label, CAEM just simply assigns that label to the new instance. If the rules are not consistent in class labels, CAEM divides the rules into groups according to class labels, and all rules in the group share the same class label. Each group has a distinct label.

First, we use the Laplace expected error estimate [12] and the support of rule to estimate the accuracy of rule, called rule-strength. The Laplace expected error estimate of a rule r is defined as follows:

$$laplace\_accuracy(r) = \frac{n_c + 1}{n_{tot} + k}, \tag{3.1}$$

where k is the number of classes, n_{tot} is the total number of instances satisfying the rule's body, among which n_c instances belong to c , the predicted class of the rule.

The rule-strength of a rule r is defined as follows:

$$rule\_strength(r) = laplace\_accuracy(r) \times n_c, \tag{3.2}$$

where n_c is the number of instances which satisfy rule r.

Second, we count the strength of the group rules R{r_1, r_2, … , r_n} by calculate its average strength , called average-strength(R). It is defined as follows:

$$average\_strength(R) = \frac{\sum_{k=1}^{n} rule\_strength(r_k)}{n}, \quad (3.3)$$

where n is the number of the R.

Finally, CAEM assigns the class label of the group with maximum average-strength to the new instance.

4. Experiments

To evaluate the accuracy of CAEM, for each data set, we designed three groups of experiments. There are performed with attribute-value pair entropy measure, with attribute-value pair deviation measure and with the effect measure which integrates entropy and deviation. In the rule generation algorithm,λ is set to 0.5, max-entropy is set to 1, min-deviation is set to 0, min-confidence is set to 0.8, min-effect is set to 0.01, δ is set to 0.01, α is set to 0.6, β is set to 0.18.

4.1. Experiments of the Binary Class

We tested our algorithm for the binary class on the mushroom data set. In each group experiment, we let the train data set grows from 100 to 1000 with the step 100. One train set and ten test sets are used for each step. The results are given as the average of accuracy, number of rules and run time. All reports of the runtime only include the runtime in rule generation. The results show in Tables 8.

Table 8. Binary Class Accuracy Comparison

size	entropy	deviation	Effect	
			λ =0.5	top
100	0.9773	0.9967	0.978	0.9967
200	0.99	0.996	0.996	0.996
300	0.9907	0.9907	0.9947	0.9967
400	0.9933	0.9933	0.9927	0.994
500	0.994	0.9927	0.9947	0.9947
600	0.9873	0.9773	1	1
700	0.9953	0.9927	0.9953	0.9953
800	0.9953	0.9927	0.9953	0.9953
900	0.9953	0.9947	0.9953	0.9953
1000	0.994	0.9907	0.9987	0.9987

4.2. Experiments of the Multi-class

We tested our algorithm for the multi-class on the nursery data set, and let the data set grows from 100 to 1000 with the step 100. A 10-fold cross-validation is performed for each step. The results are given as the average of accuracy, number of rules and run time. All reports of the runtime only include the runtime in rule generation. The results are shown as Tables 9.

Table 9. Multi-class Accuracy Comparison

size	entropy	deviation	effect	
			λ =0.5	top
100	0.72	0.73	0.73	0.73
200	0.77	0.78	0.765	0.78
300	0.7833	0.7833	0.7867	0.7867
400	0.8	0.78	0.7975	0.8
500	0.8	0.776	0.804	0.804
600	0.815	0.7883	0.8117	0.8167
700	0.8143	0.8014	0.8129	0.8157
800	0.8075	0.8012	0.8112	0.8125
900	0.8044	0.7878	0.8056	0.8078
1000	0.804	0.784	0.803	0.81

The column of top is the highest accuracy, withλ varying from zero to one. Seen from the Table 8 and Table 9, the accuracy of the multi-measure is higher single measure.

5. Conclusions

Rule-based classification algorithms perform well in classifying categorical data. However, many rule-based classifiers suffer from selecting the best attribute-value pair. So, we propose a new multi-measure, effect. It has two major features: (1) In comparison with single measure, effect can greatly decrease the number of attribute-value pair with same best values. (2) Using the effect, it is easy to distinguish which attribute-value pair's bias toward the same class label. Our experiments show that multi measure achieves higher classification accuracy than single measure.

Acknowledgements

This work is funded by China NSF program (Nos. 61170129), a grant from the Foundation of Minnan Normal University (No. SK08001).

References

[1] J. R. Quinlan, "C4.5: programs for machine learning", Morgan Kaufmann, **(1993)**.
[2] W. Zhu, "Relationship among basic concepts in covering-based rough sets", Information Sciences, vol. 14, no. 2478, **(2009)**.
[3] N. Senan, R. Ibrahim, N. M. Nawi, I. T. R. Yanto and T. Herawan, "Rough Set Approach for Attributes Selection of Traditional Malay Musical Instruments Sounds Classification", IJDTA, vol. 4, **(2011)** September, pp. 59-76.
[4] J. T. Yao and Y. Y. Yao, "Induction of classification rules by granular computing", Rough Sets and Current Trends in Computing, vol. 3, no. 331, **(2002)**.
[5] B. Liu, W. Hsu and Y. Ma, "Integrating classification and association rule mining", Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, New York, America, **(1998)** August 27-31.
[6] G. Dong, X. Zhang, L. Wong and J. Li, "CAEP: classification by aggregating emerging patterns", Proceedings of the Second International Conference on Discovery Science, Tokyo, Japan, **(1999)** December 6-8.
[7] W. Li, J. Han and J. Pei, "CMAR: accurate and efficient classification based on multiple class-association rules", The 2001 IEEE International Conference on Data Mining, California, USA, **(2001)** November 29-December 2.

[8] T. Cover and P. Hart, "Nearest neighbor pattern classification", IEEE Transactions on Information Theory, vol. 13, no. 21, **(1967)**.
[9] B. Dasarathy, "Nearest Neighbor Pattern Classification Techniques", IEEE Computer Society Press, Los Alamitos, **(1991)**.
[10] S. W. Purnami, J. M. Zain and T. Heriawan, "An alternative algorithm for classification large categorical dataset: k-mode clustering reduced support vector machine", IJDTA, vol. 4, **(2011)** March, pp. 19-30.
[11] N. M. Nawi, N. A. Hamid, R. S. Ransing, R. Ghazali and M. N. M. Salleh, "Enhancing Back Propagation Neural Network Algorithm with Adaptive gain on Classification Problems", IJDTA, vol. 4, **(2011)** June, pp. 65-76.
[12] X. Yin and J. Han, "CPAR: classification based on predictive association rules", Proceedings of the Third SIAM International Conference on Data Mining, San Francisco, CA, USA, **(2003)** May 1-3.
[13] J. R. Quinlan and R. Cameron-Jones, "FOIL: A midterm report. European Conference on Machine Learning", Vienna, Austria, **(1993)** April 5-7.
[14] W. Cohen, "Fast effective rule induction", Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, **(1995)** July 9-12.
[15] J. Cendrowska, "PRISM: An algorithm for inducing modular rules", International Journal of Man-Machine Studies, vol. 27, no. 349, **(1987)**.
[16] R. C. Boswell, "Rule induction with CN2: some recent improvements", European Working Session on Learning, Porto, Portugal, **(1991)** March 6-8.

Authors

Tianzhong He received the MSc degree in Computer Applications Technology from Nanchang University in 2005 and is a member of CCF in China. Now he is a lecturer at Department of Computer Science and Engineering, Minnan Normal University, Zhangzhou, Fujian, China. His major field of study is datamining.

Zaixiang Huang received the MSc degree in Computer Applications Technology from ZhongNan University in 2005 and is a member of CCF in China. Now he is a lecturer at Department of Computer Science and Engineering, Minnan Normal University, Zhangzhou, Fujian, China. His major field of study is datamining.

Zhongmei Zhou, Doctor, Professor, Master Tutor, teacher of Department of Computer Science and Engineering, Minnan Normal University, Zhangzhou, Fujian, China. His major field of study is datamining.