Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 984375, 6 pages
http://dx.doi.org/10.1155/2014/984375

Research Article

Classification Based on Pruning and Double Covered Rule Sets for the Internet of Things Applications

Shasha Li, Zhongmei Zhou, and Weiping Wang

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

Correspondence should be addressed to Shasha Li; shasha6320@163.com

Received 14 October 2013; Accepted 25 November 2013; Published 5 January 2014

Academic Editors: W. Sun, G. Zhang, and J. Zhou

The Internet of things (IOT) is a hot issue in recent years. It accumulates large amounts of data by IOT users, which is a great challenge to mining useful knowledge from IOT. Classification is an effective strategy which can predict the need of users in IOT. However, many traditional rule-based classifiers cannot guarantee that all instances can be covered by at least two classification rules. Thus, these algorithms cannot achieve high accuracy in some datasets. In this paper, we propose a new rule-based classification, CDCR-P (Classification based on the Pruning and Double Covered Rule sets). CDCR-P can induce two different rule sets A and B. Every instance in training set can be covered by at least one rule not only in rule set A, but also in rule set B. In order to improve the quality of rule set B, we take measure to prune the length of rules in rule set B. Our experimental results indicate that, CDCR-P not only is feasible, but also it can achieve high accuracy.

1. Introduction

The Internet of things is one of the hot topics in recent years. It has integrated many kinds of modern technology. By these kinds of technology, it produces large-scale data in IOT. In order to handle these large data, it requires techniques and methods of data mining and machine learning [1–6].

As one of the most important tasks of data mining, classification has been widely applied in IOT. The main idea of classification is that builds classification rules. According these rules, we can predict the class label for unknown objects.

Traditional rule-based classifications usually use greedy approach, such as FOIL [7], CPAR [8], and CMER [9]. These methods repeatedly search for the current best one rule or best-k rules and remove examples covered by the rules. They cannot guarantee that all instances can be covered by at least two classification rules. As a result, some traditional classifiers have less classification rules. Their accuracy may not be high. Decision tree classifiers produce classification rules by constructing classification trees, such as ID3 [10], C4.5 [11], and TASC [12]. The process of building a decision tree does not need to delete any examples. All examples can find only one matching rule in the classification rule set. That is why decision trees often generate small rule sets and cannot achieve high accuracy in some data.

Aiming at these weaknesses, we propose a novel Double Covered Rule sets classifier called CDCR-P (Classification based on the Pruning and Double Covered Rule sets). CDCR-P generates two different rule sets A and B and then prunes the rule set B. Each instance can be covered by at least one rule from rule set A. At the same time, each instance can be covered by at least one rule from rule set B. CDCR-P has four aspects. First, CDCR-P generates rule set A. We select several best values which can just cover the training set to construct a candidate set. CDCR-P employs candidate set to produce rule set A. Second, in order to induce rule set B, we remove the values of candidate set in training data and select other several best values to induce rule set B. Rule set A is fully different from rule set B. Third, each instance can find at least two matching rules. One of the rules is from rule set A, and another is from rule set B. Forth, we prune the length of rules in rule set B, so as to improve the quality of rule set B. Our method has the following advantages.

(1) CDCR-P can produce two rule sets. Thus, CDCR-P can generate large number of classification rules.

```
Input: Training data T = {t_1, t_2, ..., t_n}
Output: T_1, candidate value CV_1
Method:
(1) T_1 = φ, CV_1 = φ, Length = 0;
(2) compute the information gain of each sample s;
(3) sort s according to the information gain in descending order. If two sample have the
    same information gain, sort the two sample according to the support;
(4) Length = T . Count × 1/3;
(5) While T ≠ φ
(6)     S = S − {a}, where a is the first element of S;
(7)     for each tuple t in T
(8)         While t . Contains(a) && T_1 . Count <= Length
(9)             T_1 = T_1 ∪ {t};
(10)            CV_1 = CV_1 ∪ {a};
(11)            T = T − {t};
(12)        end while
(13)    end for
(14) end while
(15) return T_1, CV_1
```

Algorithm 1: Dividing training set into three smalldatasets.

(2) All instances in training set can be matched by at least two classification rules.

(3) CDCR-P can achieve high accuracy by combining rule set A with rule set B.

The paper is organized as follows. In Section 2, we introduce the method of CVCR (Classification based on Value Covered Rules). In Section 3, we propose a new classifier CDCR-P and discuss how to use CDCR-P to classify new objects. We report our experimental results in Section 4. We finally conclude our study in Section 5.

2. Classification Based on Value Covered Rules

In this section, we introduce the method of value covered classifier; this method is called CVCR (Classification based on Value Covered Rules).

Suppose $T = \{t_1, t_2, \ldots, t_n\}$ is a set of tuples. Each tuple t has m attributes $\{A_1, A_2, \ldots, A_m\}$. Let C be a finite set of class labels $\{C_1, C_2, \ldots, C_k\}$ and S be a set consisting of s data samples. A rule r consists of several samples s and a class label c, which takes the form of $s_1 \wedge s_2 \wedge \cdots \wedge s_l \rightarrow c$. One rule set is formed by a lot of rules which are extracted from one classifier. If tuple t satisfies $s_1 \wedge s_2 \wedge \cdots \wedge s_l$ from rule r, the t is matched by r. r predicts that t belongs to class c.

Definition 1 (information gain). Let s_i be the number of samples of S in class C_i. The information gain of an attribute value is denoted by $I(s_1, s_2, \ldots, s_k)$ and is defined as follows:

$$I(s_1, s_2, \ldots, s_k) = -\sum_{i=1}^{k} p_i \log_2(p_i), \quad (1)$$

where p_i is the probability that a literal belongs to class label c_i. p_i is estimated by s_i/s.

CVCR finds a set of values which can cover all the training set. The process of constructing CVCR is as follows.

First, CVCR sorts all literals according to the information gain in a descending order and selects several best attribute values $v_1, v_2, \ldots, v_i$ which can just cover the training set T. $v_1, v_2, \ldots, v_i$ construct a candidate set. Let these values split T to subdatasets $t_1, t_2, \ldots, t_i$, respectively. Second, CVCR connects v_i with attribute values $v_{i1}, v_{i2}, \ldots, v_{ij}$ which can just cover dataset t_i to produce patterns. Finally, repeat the above steps until the information gain of each pattern is equal to 0.

The experimental results of CVCR are shown in Table 2. The experimental results show that CVCR can achieve higher accuracy than ID3 and FOIL. Because CVCR contains the global optimal attribute values, CVCR is more feasible than ID3. However, CVCR still produces less classification rules, which cannot guarantee that each instance can be matched by at least two rules.

3. Classification Based on Pruning and Double Covered Rule Sets

In this section, we produce a new method CDCR-P. First, we show the process of how to induce rule sets A and B. Second, we describe the method of how to prune rule set B. Finally, we give the way of how to use the two rule sets A and B to classify new objects.

3.1. Constructing Rule Sets A and B. Based on the idea of CVCR, we continue mining knowledge in-depth. This approach divides the training set T into three small datasets T_1, T_2, and T_3 according to candidate set. The method contains four steps. Step 1, we select several best attribute values $v_1, v_2, \ldots, v_k$ from candidate set which can just cover one-third of training set. Attribute values $v_1, v_2, \ldots, v_k$ have less information gain. The tuples which contain one of $v_1, v_2, \ldots, v_k$ form the small dataset T_1. The process is shown as Algorithm 1. We form T_2, T_3 using the same way as T_1.

```
Input: Dataset D
Output: A set of cover value cv
Method: find a set of cover value cv that can cover D
(1) data set d = D; cover value cv = ϕ
(2) compute the information gain of each sample s in D;
(3) sort s in according to the information gain in descending order. If two sample have
      same information gain, sort the two sample according to the support;
(4) while d ≠ ϕ
(5)     S = S − {a}; where a is the first element of S;
(6)     cv = cv ∪ {a};
(7)     remove from d that all examples contain a;
(8) end while
(9) return cv
```

Algorithm 2: Finding covered values in dataset.

```
Input: Training data T_i, CV_i
Output: Rule set A
Method:
(1) Rule set R = ϕ, dataset = ϕ, cover value cv = ϕ, IniteQueue Q;
(2) while CV_i = ϕ
(3)     Q . push(a) where a is first element of CV_i;
(4)     CV_i = CV_i − {a};
(5) end while
(6) while !Q . empty()
(7)     pattern x = Q . front(); Q . pop();
(8)     if x . length > max _rule_length continue;
(9)     compute the information gain of x;
(10)    if x . information gain == 0
(11)        R = R ∪ {x};
(12)    else
(13)        dataset = dataset ∪ each t . contains(x);
(14)        found cv in dataset;
(15)        connect x with cv;
(16)        Q . push();
(17)        dataset = ϕ, cv = ϕ;
(18)    end if
(19) end while
(20) return R
```

Algorithm 3: Inducing rule set A.

Step 2, according to $v_1, v_2, \ldots, v_k$, T_1 is split into datasets $t_1, t_2, \ldots, t_n$. We find cv_i (a set of cover values) from t_i on the basis of information gain. The measure of cv_i is the same as CVCR, shown as Algorithm 2. CDCR connects v_i with cv_i to produce patterns. If the information gain of pattern is equal to 0, $X \rightarrow C$ belongs to rule set A, shown as Algorithm 3. Step 3, CDCR recalculates the CV (cover values) in T_1 excluding $v_1, v_2, \ldots, v_k$. CV splits T_1 into some datasets and connects covered value in each dataset to produce new rules. These rules belong to rule set B, shown as Algorithm 4. Finally, we remove T_1 from T and iterate the process until T_2, T_3 are trained. Rule set A is the same as CVCR. Both rule sets A and B belong to CDCR.

3.2. Pruning Rule Set B. In order to improve the quality of rule set B, we introduce a new method CDCR-P (Classification based on the Pruning and Double Covered Rule sets).

Definition 2 (confidence). The confidence of sample X is defined as follows:

$$\text{conf}(X) = \frac{\text{count}(X_c)}{\text{count}(X)} \times 100\%, \tag{2}$$

where $\text{count}(X_c)$ means the number of tuples which contain sample X in class c.

The confidence of rules that CDCR generated is equal to 100%. We modify the length of rule set B. The rules are

```
Input: Training data T_i, CV_i
Output: Rule set B
Method:
(1) Rule set R = φ, dataset = φ, cover value cv = φ, candidate set cs = φ, IniteQueue Q;
(2) add cover value which can cover T_i to cs, and cs ∉ CV_i
(3) while cs ≠ φ
(4)       Q . push(a) where a is first element of cs;
(5)       cs = cs − {a};
(6) end while
(7) while !Q . empty()
(8)        x = Q . front(); Q . pop()
(9)        If x . length > max _rule_length continue;
(10)      compute the information gain of x;
(11)      if x . information gain == 0
(12)         R = R ∪ {x};
(13)      else
(14)         dataset = dataset ∪ each t . Contains(x);
(15)         Found cv in dataset, and cv ∉ CV_i;
(16)         connect x with cv;
(17)         Q . push();
(18)         dataset = φ, cv = φ;
(19)      end if
(20) end while
(21) return R
```

Algorithm 4: Inducing rule set B.

generated when the confidence is 100% in the small dataset T_i instead of in the whole training set T. Each rule is marked with the confidence in T. Thus, rule set B in CDCR-P is shorter than rule set B in CDCR.

3.3. Classifying Unknown Examples. In this part, we give the method of how to use CDCR and CDCR-P to classify unknown instances.

Definition 3 (support). The support of sample X is denoted by

$$\sup(X) = \frac{\text{count}(X)}{|T|} \times 100\%, \tag{3}$$

where count(X) means the number of tuples which contain sample X. $|T|$ is the number of tuples in training data.

When testing unknown examples, CDCR selects the matched rule with the highest support. If some rules have the same support, we select the maximum number of matched rules in each class.

CDCR-P first considers the rule with the highest confidence. If two rules have the same confidence, CDCR-P sorts the two rules according to the support.

Definition 4 (missing match rate). If the test instance cannot find any match rule, this unclassified instance is considered mismatch. The missing match rate is defined as

$$\frac{\text{count (unclassified instance)}}{|T|} \times 100\%, \tag{4}$$

where count (unclassified instance) means the number of tuples which cannot be matched by rules.

Table 1: Characteristics of UCI datasets.

Dataset	No. of instances	No. of attributes	No. of classes
Balance	625	5	3
Breast	699	10	2
Car	1728	7	4
Lymph	148	18	4
Monks	432	7	2
Mushroom	400	23	2
Soybean	307	36	19
SPECT	267	23	2
Tic-tac	958	10	2
Zoo	101	16	7
Cleve	303	13	2
Heart	270	13	2
Iris	150	4	3
Wine	178	14	3

4. Experiments

We show the experimental results in 14 UCI datasets. The character of each data is shown in Table 1. All the experiments are performed on a 2.2 GHz PC with 2.84 G main memory, running Microsoft Windows XP. Experiments run tenfold cross validation method for each data.

Table 2: The accuracy of ID3, FOIL, CVCR, CDCR, and CDCR-P.

Dataset	ID3	FOIL	CVCR	CDCR	CDCR-P
Balance	0.3716	0.4929	0.7810	0.7985	0.8289
Breast	0.9042	0.9342	0.9571	0.9556	0.9585
Car	0.7298	0.7714	0.8837	0.9248	0.8767
Lymph	0.7148	0.7424	0.8181	0.81	0.81
Monks	0.9448	0.8146	0.8959	0.9514	0.9537
Mushroom	0.985	0.995	0.99	0.99	0.99
Soybean	0.4102	0.4172	0.8180	0.8276	0.8601
SPECT	0.7181	0.752	0.7303	0.7453	0.8053
Tic-tac	0.8215	0.9875	0.8684	0.9541	0.9530
Zoo	0.97	0.9409	0.9709	0.9609	0.8918
Cleve	0.7426	0.7423	0.8152	0.8216	0.8482
Heart	0.8148	0.8148	0.7556	0.7852	0.7963
Iris	0.7733	0.9533	0.8133	0.8133	0.8533
Wine	0.96	0.9379	0.983	0.9712	0.9882
Average	0.7758	0.8069	0.8629	0.8793	0.8867

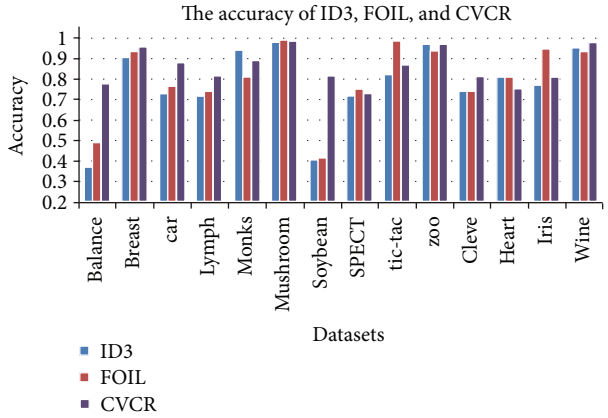

Figure 1: The accuracy of ID3, FOIL, and CVCR.

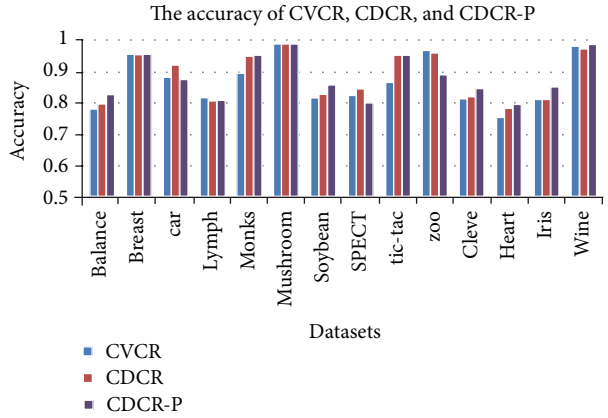

Figure 2: The accuracy of CVCR, CDCR, and CDCR-P.

In Table 2, we give the accuracy of ID3, FOIL, CVCR, CDCR, and CDCR-P. Figure 1 gives the accuracy of ID3, FOIL, and CVCR. CVCR employs the idea of covered values; these cover values are the global optimal attribute values in training data. From Figure 1 and Table 2 we can see that CVCR can achieve higher accuracy than ID3 and FOIL. Figure 2 gives the accuracy of CVCR, CDCR, and CDCR-P. CDCR not only uses the method of covered values, but also produces two rule sets A and B. Each instance can be matched at least by one rule from rule set A and rule set B. From Figure 2 and Table 2 we can see that CDCR can achieve higher accuracy than CVCR. Based on all advantages of CVCR and CDCR, CDCR-P take measure to prune the length of rule set B. The experimental results show that CDCR-P has the highest accuracy.

Table 3 displays the missing match rate of ID3, FOIL, CVCR, CDCR, and CDCR-P. CVCR can produce more rules than ID3 and FOIL. From Table 3 we can see that the missing match rate is decreased obviously by CVCR. CDCR produces two rule sets. Therefore, CDCR produces more rules than CVCR. From Table 3 we can see that the missing match rate of CDCR is lower than CVCR. CDCR-P modifies the length of rule set B; the quality of rules in CDCR-P is higher than CDCR. The experiments indicate that the mismatch rate of CDCR-P is the lowest.

Through all the above experimental results, we can conclude the following. (1) It is necessary for us to construct two rule sets. (2) It is necessary to prune rule set B. (3) CDCR-P can achieve high accuracy and has an excellent result in missing match rate.

5. Conclusions

Classification has been widely applied in IOT. The accuracy of classification is an important factor in classification task. The traditional rule-based classifications cannot guarantee that all test cases can be matched by two rules. They usually generate less classification rules. Thus, the accuracy of these algorithms may be low in some data. In this paper, a novel approach CDCR-P is proposed. CDCR-P generates two rule

Table 3: The missing match rate of ID3, FOIL, CVCR, CDCR, and CDCR-P.

Dataset	ID3	FOIL	CVCR	CDCR	CDCR-P
Balance	0.422	0.3518	0.0910	0.0799	0.0016
Breast	0.0529	0.0443	0.0029	0	0
Car	0.2165	0.1945	0.0231	0.0156	0
Lymph	0.0624	0.1014	0.0267	0	0
Monks	0	0.1023	0.0486	0.0486	0
Mushroom	0.005	0.005	0	0	0
Soybean	0.0559	0.2787	0.0097	0	0
SPECT	0.0566	0.0634	0.1387	0.1313	0
Tic-tac	0.0366	0.0094	0.0052	0	0
Zoo	0.01	0.0591	0	0	0
Cleve	0.0231	0.0795	0.0298	0.0265	0
Heart	0	0.0148	0.0704	0.0444	0.0111
Iris	0.1733	0.0067	0.1533	0.1533	0.06
Wine	0	0.0454	0.0056	0	0
Average	0.0796	0.0969	0.0432	0.0357	0.0052

sets: rule set A and rule set B. All instances can be matched by at least one rule not only in rule set A, but also in rule set B. This method greatly increases the number of extracted rules. Thus, it gets more information from training data. Our experimental results show that the methods of CDCR-P can produce more rules and achieve high accuracy. In future research, we will perform an in-depth study on combining distributed data mining with IOT in order to improve the efficiency of CDCR-P.

Conflict of Interests

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

Acknowledgments

This work is funded by China NFS program (no. 61170129), Fujian Province NSF Program (no. 2013J01259), and Minnan Normal University Postgraduate Education Project (no. 1300-1314).

References

[1] 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*, pp. 250–258, 2013.

[2] M. A. Feki, F. Kawsar, M. Boussard, and L. Trappeniers, "The internet of things: the next technological revolution," *IEEE Computer*, vol. 46, no. 2, pp. 24–25, 2013.

[3] W. Zhu, J. Yu, and T. Wang, "A Security and privacy model for mobile RFID systems in the internet of things," in *Proceedings of the IEEE 14th International Conference on Communication Technology (ICCT '12)*, pp. 726–732, November 2012.

[4] C. K. Tham and T. Luo, "Sensing-driven energy purchasing in smart grid cyber-physical system," *IEEE Transactions in Systems, Man and Cybernetics A*, vol. 43, no. 4, pp. 773–784, 2013.

[5] X. Xing, J. Wang, and M. Li, "Services and key technologies of the internet of things," *ZTE Communications*, no. 2, pp. 26–29, 2010.

[6] 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.

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

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

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

[10] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[11] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Series in Machine Learning, Kaufmann Academic, 1993.

[12] Y.-L. Chen, W.-H. Hsu, and Y.-H. Lee, "TASC: two-attribute-set clustering through decision tree construction," *European Journal of Operational Research*, vol. 174, no. 2, pp. 930–944, 2006.