A Novel Text Classification Approach Based on Enhanced Association Rule

Jiangtao Qiu[1], Changjie Tang[1], Tao Zeng[2], Shaojie Qiao[1], Jie Zuo[1], Peng Chen[1], and Jun Zhu[3]

[1] School of Computer Science, Sichuan University, Chengdu, China
[2] Computer and Information Engineering College, Tianjin Normal University, Tianjin, China
[3] National Center for Birth Defects Monitoring, Chengdu, China
qjt163@163.com,tangchangjie@cs.scu.edu.cn

Abstract. The current research on association rule based text classification neglected several key problems. First, weights of elements in profile vectors may have much impact on generating classification rules. Second, traditional association rule lacks semantics. Increasing semantic of association rule may help to improve the classification accuracy. Focusing on the above problems, we propose a new classification approach. This approach include: (1) Mining frequent item-sets on item-weighted transactions; (2) Generating enhanced association rule that has richer semantics than traditional association rule. Experiments show that new approach outperforms CMAR, S-EM and NB algorithms on classification accuracy.

Keywords: Association Rule, Text Classification, Enhanced Association Rule.

1 Introduction

Text classification is an important field on data mining and machine learning. Many studies on text classification have been conducted in the past. Existing techniques include SVM, K-NN, ANN, Naïve Bayes and etc. Association rule based classification is a new classification approach. In [2,9,10,11], association classification has been proved to have higher classification accuracy than other approaches.

Generally, designing an association rules based text classifier includes the following steps: (a) Extract profile vectors of texts. (b) Mine frequent item-sets from profile vectors and generate association rules. (c) Prune rules, and then build text classifier. To build high accurate classifier, however, two important problems must be thought out. First, weights of elements in profile vectors may have much impact on mining classification rules. Second, traditional association rule lacks semantics. Increasing the semantics of association rules may help to improve the classification accuracy.

Focusing on the above problems, we propose a new text classification approach based on enhanced association rule. Main contributions are summarized as follow. (i) Regarding profile vector as item-weighted transaction, an algorithm is proposed to

R. Alhajj et al. (Eds.): ADMA 2007, LNAI 4632, pp. 252–263, 2007.

mine frequent item-sets on item-weighted transaction. The algorithm highlights those items having large weight. (ii) A method is proposed to generate enhanced association rule that have richer semantic than traditional association rule. Experiments show that enhanced association rule based text classifier may improve the classification accuracy apparently.

The remaining of this paper is organized as follows. Section 2 gives a brief introduction to related works. Section 3 revisits the description of the problems. Section 4 gives an extensive introduction to our works on building enhanced association based text classifier. Section 5 proposes an algorithm for enhanced association rule based text classification. Section 6 gives a thorough study on classification performance in comparison with other algorithms. Section 7 summarizes our study.

2 Related Works

CBA (Classification Based on Association rule)[2] algorithm first generates candidate rules satisfied supports and confidence threshold, and then prunes these rules by choosing rules with the highest confidence. Experiments show that CBA has better performance than C4.5. CMAR[9] employed a CR-Tree to mine association rules. Then CMAR uses a weighted χ^2 analysis and multiple strong association rules to perform classification. CPAR[10] directly generates classification rules from training set without generating candidate rules. The above algorithms do not take aim at text classification. Thus they do not involve preprocessing of profile vectors, but focus on pruning rules. For the text classification, training set is text collection. Therefore preprocessing of profile vectors must be thought out carefully.

The researchers have proposed some frequent item-sets mining methods on item-weighted transactions. In [6], author defines a domain for each item's weight, and then use two steps to generate weighted association rules. First, ignores the weight and uses traditional algorithm to find frequent item-sets. Second, searches frequent item-sets that satisfy support, confidence and density threshold by space partition. In [7], author gives a fixed weight to each item, and constructs the weighted FP-Tree. In [11], author mine frequent item-sets with weighted item from texts collection. However, the above methods are not suitable for mining frequent item-sets from text profile vectors.

Predicate Association Rule, which is defined and studied in [8], is a kind of enhanced association rule. Compared with traditional association rule, predicate association rule adds more logic connectives in the proposition formula. Thus predicate association rule has richer semantics than traditional association rules. With its advantages, we use predicate association rule to build text classifier so that better classification performance can be reached.

3 Problems Description

3.1 Predicate Association Rule

Let $I=\{I_1, I_2,..., I_n\}$ be a set of variables. X and Y be two subsets of I ($X \subset I$, $Y \subset I$, and $X \neq \varnothing$, $Y \neq \varnothing$, $X \cap Y = \varnothing$). A Traditional Association Rule (TAR hereafter) is an

implication expression $P \rightarrow Q$. P is a proposition formula made of variable $I_i \in X$ and conjunction $\wedge$; Q is a proposition formula made of variables in Y and conjunction $\wedge$. For example, $I_2 \wedge I_3 \wedge I_4 \rightarrow I_7 \wedge I_9$ is a TAR.

Predicate Association Rule (PAR hereafter), as a kind of enhanced association rule, was defined in [8]. PAR adds conjunction $\wedge$, disjunction $\vee$ and negation $\neg$ in the proposition formula. Thus PAR has richer semantics and stronger expressing strength than TAR.

Let $I=\{I_1, I_2,\ldots,I_n\}$ be a set of variables, $K=\{\wedge, \vee, \neg\}$ be a set of logic connectives and f $(I_1, I_2,\ldots, I_n)$ be a proposition formula that consists of I and K. Then PAR is an implication expression $P \rightarrow Q$ (P=f(X), $X \subset I$, Q=f(Y), $Y \subset I$, $X \neq \varnothing$, $Y \neq \varnothing$, $X \cap Y = \varnothing$). For example, $(I_2 \wedge I_3) \vee (\neg I_4) \rightarrow I_7 \wedge I_9$ is a PAR.

3.2 Association Rule Based Text Classification

Let $S=\{s_1,\ldots, s_j,\ldots, s_n\}$ be a collection of texts. Each text has a pattern $(A_1,\ldots, A_k)$ that is a collection of attribute-value. In the association rule (called as classification rule while association rule are used for classification) $P \rightarrow c$, P is a proposition formula that consists of attribute-values and logic connective and c is a class label. Association rules based text classifier is a function that maps a pattern $(A_1,\ldots,A_k)$ to a class label.

Given a text object s and its pattern $(A_1,\ldots,A_k)$, if there exists one classification rule r: $P \rightarrow c$ in classifier TC(c) and P is subset of pattern of s, $P \subseteq \{A_1,\ldots, A_k\}$, we call rule r **recognize** sample s, and then s will be mapped into c. If there exists no rule that recognize s, s will be mapped into $\neg c$.

3.3 Mining Frequent Item-Set on Item-Weighted Transactions

In our study, a profile vector will be regarded as an item-weighted transaction.

Definition 1 (item-weighted transaction): Let $I=\{I_1,\ldots, I_i\}$ be a set of items, w_i be weight of I_i. A 2-tuple $<I_i, w_i>$ is called weighted item. An item-weighted transaction, IWT for short, is a set of weighted items $T=\{<I_1, w_1>,\ldots,<I_i, w_i>\}$.

Intuitively, IWT emphasize effect of items by their weight. *frequent item-sets* mined from IWTs should highlight items that have large weights.

Obviously, traditional definitions of *support* and *frequent item-set* are not suitable to generate frequent item-sets on item-weighted transactions. Therefore, we propose new definitions.

Definition 2 (support of item-set): Let $A=\{<I_1, w_1>,\ldots,<I_i, w_i>\}$ be an item-weighted item-set. *support* of A is $MIN(w_{1,\ldots,} w_i)$.

Definition 3 (frequent item-set): Let $L=\{<I_1, w_1>,\ldots,<I_n, w_n>\}$ be the set of all items in a IWT database D. Weight of item I_i, w_i, in L is the sum of weights of the I_i in D. *MaxWeight* is $MAX(w_{1,\ldots,} w_n)$, *th* is support threshold, and then minimum support $Sup=MaxWeight \times th$. An item-weighted item-set A is called *frequent item-set* when *support* of A is not less than *Sup*.

The **Example 1** in section 4.1 may help to understand the new definitions. Obviously, traditional definition of *support* and *frequent item-set* [8] is a special case of the new definition where weight of all items in D is 1 and there is an item at least occurring in all transactions in D.

4 Designing PAR Based Text Classifier

In this study, PAR based text classifier is designed by the following steps. (1) Retrieval profile vector of texts and build VSM of training text set. (2) Use mean normalization to preprocess profile vectors. (3) Mine frequent item-sets on the set of IWT, and then generate association rules. (4) Generate PARs. (5) Prune negative rules in PARs. In Section 4.1 and 4.2, we will introduce two important parts in our works, mining frequent item-sets on IWTs and generating PARs.

4.1 Mining Frequent Item-Sets on Item-Weighted Transactions

In this study, we use VSM[3] to represent a collection of texts. Each text is regarded as a vector of terms. Terms are extracted from text in n-gram.

Frequency of each n-gram occurring in the text will be counted when extracting n-grams from a text. Then a text may be converted to a profile vector. Element of profile vector is a 2-tuple <*n-gram*,*weight*> where *weight* are computed using TFIDF[3] method. In order to reduce dimensions of profile vector, terms whose weights are smaller than a threshold will be deleted from vector.

Algorithm 1. Generating Frequent item-sets from IWTS

Input: *VSM*, Threshold *min_sup*

Output: frequent item-sets L

Method:

```
1 L1=find_frequent_1-itemsets(VSM);
2 For (k=2; Lk-1≠Ø; k++){
3   Ck=Apriori_gen (Lk-1, min_sup); // generating candidate k-itemset Ck from Ck-1
4   For each vector vec∈ VSM{ // scanning VSM
5     For each candidate c∈ Ck {
6       If subset(c, vec) // c is subset of vec
7           Add(c, vec); // weight of each item in c is added by weight of same
                         //item in vec
8       Else del (c);  //deleting c
9     }
10  }
11  Lk={c∈ Ck | c. weight≥min_sup}
12  ClearWeight (c∈ Lk)}// set weight of each item of c to zero
13 return L=⋃k Lk ;
```

According to samples in the training set belonging or not belonging to class c when we build classifier TC(c), training set will be divided into the positive training set and the negative training set. Then each profile vector in VSM is regarded as an IWT and association rules are generated by mining frequent item-sets on positive training set. However, neglecting weight of elements in IWT when mining frequent item-set will play down effect of the most representative item on classification. For example, there are two profile vectors V_1={<t_1,15>,<t_2,10>,<t_3,2>}, V_2={<t_2,3>,<t_3,1>}. Frequent item-sets whose support are not less than 2 are {t_2},{t_3} and {t_2,t_3} when neglecting weights of term. Although weight of t_1 is 15 that is greater than weight of other terms, {t_1} is not frequent item-set. This problem often is overlooked, but it must be thought about carefully when designed association rules based text classifier.

In this study, frequent item-sets are generated from IWTs, and then classification rules are derived. In order to mine frequent item-sets from IWTs, we propose Algorithm 1, the improved apriori algorithm.

Example 1. There are three profile vectors {a:15, b:4, c:1}, {a:1, c:1}, {b:3, c:2}. Fig.1 shows the process using Algorithm 1 to mine frequent item-sets on profile vectors. Let support threshold be th=15%. From item collection L1, the greatest weight in collection is 16. By definition 3, minimum support is 16×15%=2.4. Then frequent item-sets {a}:16, {b}:7, {c}:4, {a, b}:4, {b, c}:3 may be derived.

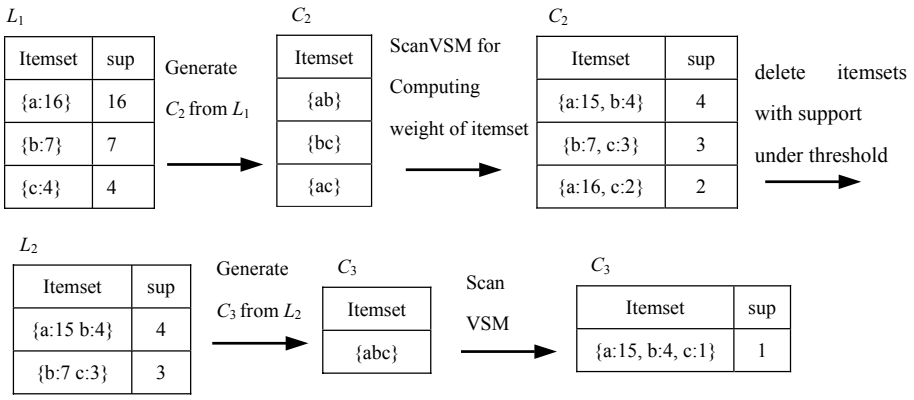

Fig. 1. Mining frequent Itemsets on feature vectors

4.2 Generate PARs

In Section 4.1, we have introduced how to generate classification rules from VSM. These rules, however, are traditional association rules and they lack semantics. Thus we need to further generate PARs based on these classification rules.

Let TC(c) be a association rule based classifier, S be training set and $S_c \subseteq S$ be training subset in which samples have class label c. $\neg S_c$, called *self* set in this paper,

is supplementary set of S_c. $\neg S_c$ represent samples in the training set that do not have class label c. If a rule r in TC(c) recognizes sample $s(s \in \neg S_c)$, we call r recognizes *self*.

According to *precision* formula in classification

$$\text{Precision}=\frac{\text{number of samples correctly labeled as positive } n}{\text{number of samples labeled as positive } m} \tag{1}$$

If classifier recognizes more *self*s, m will increase while n keep unchanged. Thus precision of classifier will decrease.

If we simply delete those rules recognizing *self*, it will not be helpful for improving classification precision. For example, a rule r wrong recognize i samples and correctly recognize j samples, $j>i$. If we delete rule r from classifier, *precision* is $(n\text{-}j)/(m\text{-}i)$ that is less than n/m. It can be a good idea for improving classification *precision* that reduces recognizing negative samples while keep recognizing positive samples unchanged.

Suppose a rule r: $P \rightarrow c$ recognizes a sample A in S_c and a sample B in $\neg S_c$. It can be concluded that P is subset of both pattern of A and pattern of B, $P \subseteq A$ and $P \subseteq B$. Because B is in $\neg S_c$, A new rule $Q \rightarrow \neg c$, $Q \subseteq B\text{-}A$, can be derived. From expression $P \rightarrow c \wedge Q \rightarrow \neg c$, we can further infer the expression $P \wedge \neg Q \rightarrow c$ in which P, $P \rightarrow c$, Q and $Q \rightarrow \neg c$ are called positive logical formula, positive rule, negative logical formula and negative rule, respectively. For sample A in S_c, $P \subseteq A$, classification rule $P \wedge \neg Q \rightarrow c$ may label c to A correctly. For sample B in $\neg S_c$, $P \subseteq B$ and $Q \subseteq B\text{-}A$, although P is subset of pattern of B, $P \subseteq B$, Q is also subset of pattern of B, $Q \subseteq B$. Thus rule $P \wedge \neg Q \rightarrow c$ will label $\neg c$ to B.

In [12], it was proved that, for texts belonging to same class, there always exists a set of n-grams occurring in these texts in high frequency. Two texts belonging to same class will have roughly same n-gram frequency distribution. Therefore, if text D and B belong to same class, they will have roughly same n-gram frequency distribution. If rule r: $P \rightarrow c$ wrong recognize B ($B \in \neg S_c$), $Q \subseteq B\text{-}P$, and then PAR r: $P \wedge \neg Q \rightarrow c$ may be generated, there will be great probability on Q being subset of pattern of text D, $Q \subseteq D$, because D and B belong to same class. Thus rule r will label $\neg c$ to text D in high probability. By the above analysis, we can conclude that PAR may reduce wrong recognized samples in $\neg S_c$.

Let E be a text belonging to class c. According to the above n-gram frequency distribution, negative logical formula of rule r, Q, is generated from text B. B and E do not share same label. Therefore, $Q \subseteq E$ is in a small probability, i.e. rule r assign label $\neg c$ to E in a small probability. By this way, PAR may assure that number of wrong labeled positive sample reduce dramatically while number of correctly labeled sample reduce slightly. Thereby classification precision is improved.

Algorithm 2. generating predicate association rule

Input: set of classification rules *Rule*, set of selfs *self*, support threshold *Supp*
Output: Predicate Association Rules *H*
Methods:

```
1  For each rule r in Rule{
2      W=null;
3      For each sample s in Self
4        If (Recog(r,s))
5          Insert(s, W);  //inserting sample recognized by r into W
6      NegRule=CreateNegRules(W, Supp); //generating negative rule of r from W
7      DelRepeateItems(NegRule);
8      Pick(NegRule) ;      //picking up less-general negative rules
9      NewRule=GenerateNewRule(r, NegRule);
10     Insert(H,NewRule);   //inserting PARs to H
11 }
```

In Algorithm 2, function GreateNegRules will mine negative rules satisfying support threshold by using method in Section 4.1. In function DelRepeateItems, if there is the intersection of items between left part of a negative rule, P_1, and a positive rule, P_2, P= $P_1 \cap P_2$, then item $I \in$ P will be deleted from P_1.

Example 2. Let r: $A_1 \wedge A_2 \rightarrow c$ be a classification rule, the collection of profile vectors in negative training set recognized by r denoted as W; $\{A_1, A_1A_2A_5A_6, A_1A_3, A_3A_4\}$ are part of frequent item-sets mined from W. After deleting items occurring in left part of rule r from the two frequent item-sets, $\{A_5A_6, A_3, A_3A_4\}$ can be derived. Picking up less-general rule $\{A_5A_6, A_3A_4\}$, we derive proposition formula $A_5 \wedge A_6$ and $A_3 \wedge A_4$. New proposition formula $\neg(A_5 \wedge A_6 \vee A_3 \wedge A_4)$ can be derived when connecting proposition formulas using disjunction $\vee$ and appending a negation $\neg$. Finally, a new PAR $A_1 \wedge A_2 \wedge \neg(A_5 \wedge A_6 \vee A_3 \wedge A_4) \rightarrow c$ is generated after adding the new proposition formula to rule r. The PAR means that class label c may be assigned to a text when profile vector of the text include elements of $\{A_1A_2\}$ and do not include $\{A_5A_6\}$ and $\{A_3A_4\}$. In other words, if a profile vector of text includes $\{A_5A_6\}$ or $\{A_3A_4\}$, class label $\neg c$ will be assigned to the text.

5 PARs Based Text Classification

From algorithm 2, we can learn that a PAR in classifier TC(c) includes two parts: positive logical formula and negative logical formula. Negative logical formula can be divided into several parts by disjunction. Positive logical formula and No.j part of negative logical formula of PAR r_i is denoted as PosRule(r_i) and NegRule(r_i^j), respectively. Both PosRule (r_i) and NegRule(r_i^j) may be reduced to the integer. Reduction of PosRule (r_i) includes the following steps. (1) Generate a set of items from all positive logical formulas in classifier, denoted as PV. (2) A unique integer number *id*, starting from 1, is assigned to each item in PV, and then items are ranked by their *id* in ascend order. (3) Let an integer be zero, for each item $A_t \in$ PosRule(r_i)

and its *id*, No.*id* bit of the integer is set to 1. Then PosRule(r_i) of a PAR may be represented by an integer, denoted as INT(PosRule(r_i)). (4) Reduced PosRule(r_i) of all PARs in TC(c) are stored in an array Pos(c). The array is the collection of positive logical formula of PARs. An element of array is a 2-tuple <INT(PosRule(r_i^j)), RID> where RID is number of a rule in TC(c).

After generating a set of items from all negative logical formulas in classifier, denoted as NV, NegRule(r_i^j) of PAR in TC(c) also may be reduced into an integer number, denoted as INT(NegRule(r_i^j)), by the above method. All reduced NegRule(r_i^j) of PARs in TC(c) are stored in an array Neg(c). The array is the collection of negative logical formula of PARs in TC(c). Structure of Neg(c) is same with Pos(c).

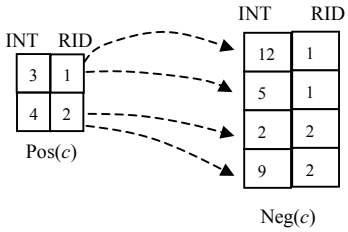

Fig. 2. Storing Reduced PARs

Example 3. There are two PARs in TC(c) r_1: $A_1 \wedge A_2 \wedge \neg(A_5 \wedge A_6 \vee A_3 \wedge A_5) \rightarrow c$ and r_2: $A_3 \wedge \neg(A_4 \vee A_3 \wedge A_6) \rightarrow c$. The set of items in positive logical formula is PV={ A_1, A_2, A_3 }; the set of items in negative logical formula is NV={ A_3, A_4, A_5, A_6 }. Items A_1, A_2, A_3 are assigned integer number 1, 2, 3 in PV, respectively. Items A_3, A_4, A_5, A_6 are assigned integer number 1,2,3,4 in NV, respectively. After PosRule(r_1)= $A_1 \wedge A_2$ is reduced, its binary integer is 011 and INT(PosRule(r_1))=3. NegRule(r_1^1)= $A_5 \wedge A_6$ also may be reduced into a binary integer 1100 and INT(NegRule(r_1^1))=12. By the same way, INT(NegRule(r_1^2))=5, INT(PosRule(r_2))=4, INT(NegRule(r_2^1))=2 and INT(NegRule(r_2^2))=9 also can be derived. Pos(c) and Neg(c) are shown in Fig.2.

An unlabeled text may get a class label by using Algorithm 3.

Algorithm 3(PARC). Predicate Association Rule Based Text Classifying

Input: unlabeled text *s*, PAR based Classifier TC(c), pruning threshold *t*
Output: label (c or $\neg c$)
Methods:

```
1  vec=GetVector(s); // extracting profile vector of s
2  vec=Prune(t, vec);//deleting elements in vec whose frequency under t
3  pv=GetPV;// getting the set of items in positive logical formula of TC(c)
4  posInt=Reduce(pv, vec);// reducing PV ∩ vec to a integer
5  nv=GetNV;// getting the set of items in negative logical formula of TC(c)
6  negInt=Reduce(nv, vec);// reducing NV ∩ vec to a integer
7  If (Match(posInt, Pos(c))) // get RID of positive rule matching with posInt
8  get posInt.RID;
9  else return ¬c;
```

```
10 If (Match(negInt, Neg(c)) and posInt.RID=negInt.RID)
11   return ¬c;
12 else return c
```

In this algorithm, function GetVector extract profile vector of the unlabeled text in n-gram. However, number of n-grams may be large. By [12], n-grams with the high frequency in a text may represent feature of the text. Thus it is rational to delete n-grams whose frequencies do not satisfy the threshold because they cannot represent feature of the text. Deleting n-grams with low frequency may help to reduce their disturbance for classification. Function Match(a, b) return matching result of a and b. Two binary integers a, b are operated in logical OR. If the result is equal to a, we call b matching a.

6 Experiments and Analysis

The experiments aim at (1) studying how Algorithm 1 improve classification performance, (2) comparing PARC with other classification algorithms, such as CMAR[9], S-EM[1] and Naïve Bayes. Our experiments use 3-fold cross-validation. *precision*(Formula 1), *recall*(Formula 2) and *accuracy*(Formula 3) measures are used in our experiments.

$$\text{Recall}=\frac{\text{number of samples labeled correctly as positive}}{\text{number of all positive samples}} \tag{2}$$

$$\text{Accuracy}=\frac{\text{number of samples labeled correctly}}{\text{number of all samples}} \tag{3}$$

6.1 Dataset and Environment

Our experiments use Chinese Web Classification Training Set CCT2006[1] as dataset. This dataset include 1200 web pages distributing in eight classes. There are 150 web pages in each class. All web pages in each class are divided into three parts. Each part includes 50 pages. In 3-fold cross-validation, each time of experiment will use two parts of pages in one class as training set and another part as test set. Because there are eight classes, our experiment builds classifiers for each class respectively. For example, building a classifier TC(c) for class c, we use training set of class c as positive training set for classifier TC(c) and use test set of class c as positive test set, use all prior two parts pages of others classes as negative training set and other parts as negative test set.

Experiments were performed on an Intel C3 1.0G PC with 512M main memory and running Windows 2000 sever. PARC and CMAR are implemented in JAVA, and S-EM and Naïve Bayes(NB) are downloaded from website[2]

[1] http://www.cwirf.org

[2] http://www.cs.uic.edu/~liub/S-EM/S-EM-download.html

6.2 Experiment

6.2.1 Test 1

Test 1 compare classification performance on two situations where profile vectors are regarded as item-weighted transaction and common transaction. We build classifiers TC(i){i=1,…,8} by generating TARs on training set of eight classes respectively, and then classifiers are tested on their own positive and negative test set. *posNum* and *negNum*, in each row in Table 1, are average number of samples recognized as positive sample on positive and negative test set in 3-fold cross-validation respectively. In each fold validation, number of positive test samples is 50 and number of negative test samples is 350. *minimum support* is set at 10 in test.

Table 1. The Comparison of mining frequent item-sets on IWT and common transaction

Class	With weight		Without weight	
	posNum	*negNum*	*posNum*	*negNum*
1	32.67	54	37	17
2	21.33	53	26	17
3	31.67	52	34	20.33
4	21.67	54	26.33	32.33
5	44.67	50	44	7.33
6	30.33	54	44	44
7	19.67	51	29.33	45.67
8	42.33	11	43	12.33
Avg	30.54	43.38	35.46	24.5

From Table 1, we can observe that classifier that is built by mining item-sets with weight may increase recognizing positive samples (35.46-30.54) /50≈10% and reduce recognizing negative samples (43.38-24.5) /350≈5.4% than classifier neglecting item weight. It can be concluded that better classification accuracy can be reached when mining frequent item-sets with weight than without weight.

6.2.2 Test 2

Test 2 compare algorithm PARC with CMAR, S-EM and NB(Naïve Bayes) on classification performance.

Table 2 lists average *precision* (*Prec*), *recall* (*Reca*) and *accuracy* (*Accu*) in 3-fold cross-invalidation on eight classes.

From table 2, we can observe that PARC has same *precision* with CARM, but 15% higher *recall* and 7% higher *accuracy* than CARM.

Table 2. The comparison of PARC, CMAR, S-EM and NB on classification performance

class	PARC			CMAR			S-EM			NB		
	Prec	*Reca*	*Accu*	*Prec*	*Reca*	*Accu*	*Prec*	*Reca*	*Accu*	*Prec*	*Reca*	*Accu*
1	0.80	0.65	0.93	0.87	0.55	0.93	0.6	0.87	0.79	0.65	0.87	0.92
2	0.87	0.5	0.93	0.93	0.41	0.92	0.26	0.81	0.7	0.35	0.7	0.81
3	0.75	0.44	0.91	0.82	0.29	0.90	0.26	0.65	0.77	0.33	0.72	0.82
4	0.68	0.51	0.91	0.67	0.32	0.89	0.49	0.73	0.84	0.59	0.73	0.89
5	0.95	0.87	0.98	0.93	0.84	0.97	0.86	0.91	0.97	0.86	0.91	0.97
6	0.82	0.78	0.95	0.57	0.7	0.89	0.62	0.89	0.92	0.62	0.89	0.92
7	0.75	0.63	0.93	0.75	0.25	0.90	0.5	0.76	0.85	0.54	0.75	0.88
8	0.86	0.83	0.96	0.94	0.61	0.95	0.63	0.91	0.92	0.72	0.92	0.94
Avg	0.81	0.65	0.94	0.81	0.5	0.92	0.53	0.82	0.85	0.58	0.81	0.89

Although S-EM and NB have 17% and 16% higher *recall* than PARC, their *precision* is 28% and 23% lower and their *accuracy* are 9% and 5% lower than PARC.

From test 2, we can conclude that PARC have better performance on text classification than CAR, CMAR, S-EM and NB in an overall evaluation.

7 Conclusions

In this paper, a new text classification approach is proposed. We highlight the effect of elements with high frequency on classification by mining frequent item-sets on item-weighted transactions. The new approach generates predicate association rules that have richer semantics than traditional association rule. Experiments show that the new approach outperforms CMAR, S-EM and NB on classification accuracy.

References

1. Liu, B., Lee, W.S., Yu, P.S., Li, X.: Partially Supervised Classification of Text Document[C]. In: Proceeding of ICML- 2002 (2002)
2. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining[C]. Proceeding of KDD'98, New York (August 1998)
3. Salton, G., Wong, A., Yang, C.S.: A Vector Space Model for Automatic Indexing [J]. Communication of the ACM (1) (1995)
4. Sebastiani, F.: Machine Learning in Automated Text Categorization [J]. ACM computing Surveys, 34(1) 11–12, 32–33 (2002)
5. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation [C]. In: Proceeding of SIGMOD'00, Dallas, TX (2000)

6. Wang, W., Yang, J., Yu, P.: Efficient mining of weighted Association Rules (WAR) [J]. IBM Research Report RC 21692(97734) (March 2000)
7. Zhang, Z., et al.: Enabling Personalization Recommendation With WeightedFP for Text information Retrieval Based on User-Focus[C]. In: Proceeding of ITCC'04 (2004)
8. Jie, Z., Changjie, T., Tianqing, Z.: Mining Predicate Association Rule by Gene Expression Programming. In: Meng, X., Su, J., Wang, Y. (eds.) WAIM 2002. LNCS, vol. 2419, Springer, Heidelberg (2002)
9. Li, W., Han, J., Pei, J.: CMAR: accurate and efficient classification based on multiple class-association rules[C]. In: Proceeding of ICDM (2001)
10. Yin, X., Han, J.: CPAR: Classification based on Predictive Association Rules[C]. In: Proceeding of International Conference on Data Mining (SDM'03) (2003)
11. Xiaoyun, C., Wei, C.: Text Categorization Based on Classification Rules Tree by Frequent Patterns [J]. Journal of Software, 17(5) (May 2006)
12. Cavnar, W.B., Trenkle, J.M.: N-Gram-Based Text Categorization. In: Proceeding of Third Annual Symposium on Document Analysis (1994)
13. Zhou, S.G., et al.: A Chinese document categorization system without dictionary support and segmentation processing [J]. Journal of Computer Research and Development, 38(7) (2001)