[A Novel Feature Selection Method for Text Classification](http://dl.acm.org/citation.cfm?id=2397010) using Association Rules and Clustering

Navid Sheydaei

Department of Computer Engineering, Isfahan University of Technology, Isfahan, Iran

Mohamad Saraee

School of Computing, Science and Engineering, University of Salford-Manchester, Manchester, UK

Azar Shahgholian

Department of Computer Engineering, Isfahan University of Technology, Isfahan, Iran

Abstract

Readability and accuracy are two important features of any good classifier. For reasons such as acceptable accuracy, rapid training and high interpretability, associative classifiers have been recently used in many categorization tasks. Although features could be very useful in text classification, both training time and the number of produced rules will increase significantly due to the high dimensionality of text documents. In this paper an association classification algorithm for text classification is proposed which includes a feature selection phase to select important features and a clustering phase based on class labels to tackle this short-coming. The experimental results from applying the proposed algorithm in comparison with the results of selected well-known classification algorithms show that our approach outperforms others both in efficiency and performance.

Keywords

Association Text Classification; Feature Selection; Text Classification

1. Introduction

With the explosive growth of information, the requirements for information acquisition and management have increased rapidly. Information can be presented in both structured and unstructured format. The task of automatically classifying a set of documents into predefined categories is called class labels in this paper. Text classification offers a better understanding of information presented as text in terms of the most frequently used terminology. Various types of applications have been reported in this domain such as genre classification [1], opinion mining [2], classification of web pages based on text [3] and spam filtering [4].

Classification algorithms can be categorized into two main groups, i) traditional algorithms such as K-Nearest Neighbours[5], Decision Trees [5], and C4.5[5], and ii) Association classification algorithms[6, 7, 10, 11, 12]. Experimental results in [6, 7] show that association classification algorithms have higher accuracy in comparison to traditional algorithms.

The overall process of association classification algorithm begins by using one of the association rule mining algorithms like Apriori [8] or Fp-Growth [9]. A data set of association rules is produced and then a small set of high quality rules is selected and finally these rules are used for prediction. Since the associative classifier is a rule-based classifier, humans can easily understand its operation, and the prediction result provides a simple and direct interpretation [10]. However, associative classification suffers from efficiency problems owing to the fact that it often generates a very large number of rules in association rule mining, and it also takes an effort to select high quality rules from among them [11].

In this paper, we have proposed an algorithm called BACA (Bit-priori Association Classification Algorithm) for classifying documents. BACA inherits the idea of CACA (Class-based Associative Classification Algorithm) [12] in using a class-based strategy to filter high quality features. It reduces the search space of frequent patterns and uses OR-Tree structure in order to synchronize the existing phases in the feature selection process. In comparison with text classification algorithms, BACA has the following advantages: (1) BACA generates a small set of high quality rules to predict class labels; (2) it has the ability to predict more than one class label for each unseen document (Multi-label classification), (3) it is independent of language in classification; (4) it uses a binary structure for storing data, producing rules and classification.

The remainder of the paper is arranged as follows: Section 2 gives an overview of the related works in association classification classifiers; in Section 3 the general aspect of Associative classifications is described; in Section 4 our proposed algorithm in text classification is discussed integrally; in Section 5 experimental results using our proposed algorithm are demonstrated and compared with the results of some known algorithms and finally Section 6 provides a conclusion for this paper.

1. Related works

Many text classifiers have been proposed in the literature such as using machine learning techniques and probabilistic models, e.g. decision trees, naïve-Bayes [5], nearest neighbours. A group of classifiers called association rule-based classifiers has recently been proposed, which performs well and its effectiveness is comparable to most well-known text classifiers [13]. Some well-known association rule-base classifiers are including CBA [7], CMAR [6], CPAR [11], HARMONY [14], MCAR[15] and CACA [12] employ different approaches in rule discovery, rule ranking, rule pruning, rule prediction and rule evaluation methods. A brief description of each of them is described as followed:

One of the first algorithms to use an association rule mining approach for classification was proposed in [7] and named CBA. It implements apriori algorithm [16] in order to discover frequent itemsets in transactional data. Once the discovery of frequent items is completed, CBA proceeds by generating rules from any frequent itemsets that passes minimum confidence. These generated rules are evaluated and a subset of these rules is selected and is used for the final classifier. Finding frequent itemsets and rules generation are implemented in two separate phases in CBA. [15]

CMAR has extended the frequent pattern mining method; FP-growth; and constructs a class distribution-associated FP-tree and attempts to mine large databases efficiently. It uses CR-tree structure to store and retrieve association rules and prune them based on confidence, correlation and database coverage. HARMONY directly mines; for each training instance; one of the highest-confidence frequent classification rules that it supports and builds the classification model from the union of these rules over the entire set of instances. CPAR seeks to combine the advantages of both associative classification and traditional rule-based classification. It inherits the basic idea of FOIL [17] in rule generation and integrates the features of associative classification in predictive rule analysis, using a ‘greedy algorithm’ to generate rules directly from training data. HARMONY employs an instance-centric rule generation framework and is guaranteed to find and include the best possible rule for each training instance [14]. However, such a one-rule-per-example principle may work adversely when handling a database with uneven distribution of class labels, because a class with many training examples may have a high prediction score on that class label, which leads to a wrong prediction [18]. The MCAR algorithm uses an intersection technique for discovering frequent itemsets. MCAR consists of two main phases: rule generation and a classifier builder. In the first phase, the training data set is scanned once to discover frequent ‘1-item’ in an itemset the combined rules are then generated to produce candidate itemsets with more items. Any rule with support larger than the minimum support is created as a ‘candidate rule’. In the second phase, rules created are used to build a classifier by considering their effectiveness on the training data set. Only rules that cover a certain number of training cases are kept in the classifier [15].

Another association classification algorithm called CACA was proposed in [12]. CACA first scans the training data set, before storing data vertically like the MCAR algorithm. Next frequency counts the frequency of each attribute value are produced and sorted into descending order according to their frequencies. All frequent ‘disjoint attribute values’ TIDs are intersected to reduce the search space of frequent patterns. A TID of a frequent attribute value holds the row numbers where these attribute values occur in the training data set. Lastly, each attribute in a class group that passes the minimum confidence, is inserted in the Ordered Rule Tree (OR-Tree) as a path from the root node and its support, confidence and class are contained within the last node in the path. CACA classifies the unseen data like the CBA algorithm. Experimental results suggested that CACA performs better with reference to accuracy and computation time than other associative algorithms on the UCI data sets [19][20].

Our algorithm is inspired by CACA, by its filtering of the search space of frequent patterns but obtains the high quality features in a different manner.

The classifiers described above are not implemented for text classification and have a general application. Some algorithms such as [9, 13, 18] have been specially proposed for text classification and are within the family of Association classification algorithms.

1. Association Classification

The overall structure of association text classification algorithms is shown in Figure 1. As Figure 1 shows, this process consists of two phases, a **training** phase and a **testing** phase. In the training phase, raw data of training documents is pre-processed. In this part, depending on need, stop words and irrelevant characters are removed and words are stemmed and indexed (step 1). By mining data from this pre-processed database, we obtain the classification rules (step 2). Owing to the very large number of classification rules, some redundant rules are filtered out and a small number of important rules are selected. This process is called **pruning** (step 3). Now we have a database of classification rules. In continuing, in the testing phase, before any activity unknown documents should be pre-processed and then converted to a pattern of words and matched to the classification rules (step 4). In the final step (step 5), according to the matched rules, class labels of unknown documents are predicted.

|  |
| --- |
|  |
| **Figure 1.** Association text classification structure |

* 1. Association Rule Problem

The problem of mining association rules over basket data was introduced in [16] by Agrawal. An example of such a rule might be that 98% of customers who purchase tyres and auto accessories also require automotive services. Finding all such rules is valuable for cross-marketing and attached mailing applications. Other applications include catalogue design, add-on sales, store layout, and customer segmentation based on buying patterns. As databases involved in such applications are very large it is therefore imperative to execute fast algorithms during thess tasks [21].

Let be a set of attributes, D be a set of different transactions where each transaction T contains a set of items such that. An association rule takes the form of where X is called *antecedent* and Y is called *consequent* of the rule; the rule means X implies Y where and.

There are two important basic measures for association rules, *support(s)* and *confidence(c)*. These are usually two predefined thresholds employed by users to drop the rules that are perceived as less interesting or useful to the user. The two thresholds are called *minimum support* and *minimum confidence* respectively. For , the support of the rule is the percentage of transactions that contain and the confidence of the rule is the ratio of the number of transactions that contain to the number that contain X [21].

* 1. Selecting high-quality features

Most text data has a high-dimensional feature space that causes it to generate a lot of association rules. There are two common approaches to generate high-quality rules. First, all the association rules are produced. Therefore, many high-order association rules are generated in the induction procedure for the associative classifier. Generally, high-order rules are more informative; hence the classification with those rules performs better. [23, 24]. In these methods computational complexity is very high [10]. Second, a small number of high quality rules have been used regardless of the order of rules (or preferring high-order rules) [25, 14]. To reduce computation time, these methods remove the redundant words and select high-quality features only.

In this paper we use a method of the second type. In our proposed method, high-frequency words are first found and then clustered based on class labels. By this means, high-quality features for each class label are found. Using these features, association rules for each class label are produced; we therefore do not need to consider all the words for each class label.

1. Proposed Method

The general approach of the proposed algorithm will be as follows. Firstly, pre-processing will be applied to documents to convert them to a structured format. In the next step, frequent items for each class label will be identified. This step follows CACA approach [12]. By this means we have the features (words) that are more effective in documents for each class label. Rules are then produced using only these effective words. In finding frequent items and creating rules, a type of Apriori algorithm called Bit-Apriori [26] is used that first, saves items in a database in binary mode and secondly with the help of binary operations attempts to find the rules. Finally unseen documents are classified using these rules.

|  |
| --- |
|  |
| **Figure 2**. Steps of proposed algorithm |

As explained above, it is clear that feature selection is at the heart of the algorithm that finds desired features based on occurrence of words in document and class labels.

* 1. Convert text to structured form

Most text corpora have a high-dimensional feature space. Since they are not relational databases, there is no predetermined length for an example record. The average number of words in a document is far less than the vocabulary size, which indicates the sparsity of word distribution in text collections. A document can be modelled statistically by treating the words that constitute the document as the events. There are two different document models; the Multivariate Bernoulli Model and the Multinomial Model [10]. The former ignores the count information of words and uses only binary information (present or not-present), while the latter includes the word count [10]. As it is known that information about the multiple occurrences of words does not yield significant additional assistance for an exact classiﬁcation [27], [23], we have chosen a combination of these two ways. This means that first we count the occurrences of the word in the document (Table 1) and then ignore the words with less than the given threshold (Table 2). The threshold is selected based on the assumption that high threshold implies few rules while low threshold implies more rules. Following a number of experiments, we set the threshold to 2 which indicated that words used more than once in a document are considered frequent. A 4-fold cross validation is used to confirm the selected threshold.

Finally we save the documents in a binary table where the rows of the table are documents and the columns are the frequent words used in each document. All existing words in the document are marked as ‘1’ in the table. As the class of each document is known, a column has been added to the table to contain the value of the class.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table 1.** The number of words' repetition in the documents | | | | | | | |
| Document | W1 | W2 | W3 | W4 | W5 | W6 | Class |
| D1 | 1 | 4 | 1 | 0 | 6 | 1 | C1 |
| D2 | 3 | 3 | 2 | 1 | 4 | 5 | C3 |
| D3 | 2 | 5 | 3 | 4 | 3 | 4 | C3 |
| D4 | 5 | 2 | 1 | 7 | 1 | 9 | C1 |
| D5 | 7 | 6 | 0 | 4 | 4 | 1 | C2 |
| D6 | 10 | 3 | 7 | 9 | 0 | 0 | C2 |
| D7 | 0 | 7 | 5 | 5 | 6 | 5 | C1 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table 2.** Binary table containing frequent items in the documents | | | | | | | | | |
| Document | W1 | W2 | W3 | W4 | W5 | W6 | C1 | C2 | C3 |
| D1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| D2 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| D3 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| D4 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| D5 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| D6 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| D7 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

* 1. Find frequent words for each class label

Since we want to produce rules that will eventually predict class labels of unseen documents, we are going to find the effective words for each class label. This means we aim to find commonly used words that are repeated in many documents and have the same class label.

To achieve this goal, it is sufficient to perform a binary 'And' operation between columns of each class (C[i]) with every column of words. Subsequently, a count of the number of '1's in each column is produced (the final row of Table 3, 4 and 5). In the new tables, every cell of the table shows frequent words that are used in different documents and have same class label. In order to clarify this fact, based on previous example in Table 2 we have seven documents and three classes namely; C1, C2 and C3. Table 3, 4 and 5 present the binary “And” of each class label and words following by counting the number of “1” in each column respectively. For example, in Table 3, we can conclude that W2 is presented in three different documents; therefore using the aforementioned approach, the frequency of W2 in class label C1 is 3.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Table 3.** Frequent words in the documents for class C1 | | | | | | |
| Document | C1&W1 | C1&W2 | C1&W3 | C1&W4 | C1&W5 | C1&W6 |
| D1 | 0 | 1 | 0 | 0 | 1 | 0 |
| D2 | 0 | 0 | 0 | 0 | 0 | 0 |
| D3 | 0 | 0 | 0 | 0 | 0 | 0 |
| D4 | 1 | 1 | 0 | 1 | 0 | 1 |
| D5 | 0 | 0 | 0 | 0 | 0 | 0 |
| D6 | 0 | 0 | 0 | 0 | 0 | 0 |
| D7 | 0 | 1 | 1 | 1 | 1 | 1 |
| Total | 1 | 3 | 1 | 2 | 2 | 2 |
| **Table 4.** Frequent words in the documents for class C2 | | | | | | |
| Document | C2&W1 | C2&W2 | C2&W3 | C2&W4 | C2&W5 | C2&W6 |
| D1 | 0 | 0 | 0 | 0 | 0 | 0 |
| D2 | 0 | 0 | 0 | 0 | 0 | 0 |
| D3 | 0 | 0 | 0 | 0 | 0 | 0 |
| D4 | 0 | 0 | 0 | 0 | 0 | 0 |
| D5 | 1 | 1 | 0 | 1 | 1 | 0 |
| D6 | 1 | 1 | 1 | 1 | 0 | 0 |
| D7 | 0 | 0 | 0 | 0 | 0 | 0 |
| Total | 2 | 2 | 1 | 2 | 1 | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Table 5.** Frequent words in the documents for class C3 | | | | | | |
| Document | C3&W1 | C3&W2 | C3&W3 | C3&W4 | C3&W5 | C3&W6 |
| D1 | 0 | 0 | 0 | 0 | 0 | 0 |
| D2 | 1 | 1 | 1 | 0 | 1 | 1 |
| D3 | 1 | 1 | 1 | 1 | 1 | 1 |
| D4 | 0 | 0 | 0 | 0 | 0 | 0 |
| D5 | 0 | 0 | 0 | 0 | 0 | 0 |
| D6 | 0 | 0 | 0 | 0 | 0 | 0 |
| D7 | 0 | 0 | 0 | 0 | 0 | 0 |
| Total | 2 | 2 | 2 | 1 | 2 | 2 |

Consequently, we should save a list of words that are effective in predicting class labels. To this purpose, if the number of occurrences of a word in documents related to a class label is more than a threshold (Min\_Document\_Repetition), then it will be considered as a frequent word for the class label. In output, the most frequent words that are used in a document with the same class label as the considered class, take the value 'True'. This stage is the second phase of feature selection. For example if Min\_Document\_Repetition=2 in our example, we have:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Table 6.** Frequent words for each class label | | | | | | |
| Class | W1 | W2 | W3 | W4 | W5 | W6 |
| C1 | 0 | 1 | 0 | 1 | 1 | 1 |
| C2 | 1 | 1 | 0 | 1 | 0 | 0 |
| C3 | 1 | 1 | 1 | 0 | 1 | 1 |
|  |  |  |  |  |  |  |

The considered constant for Min\_Document\_Repetition could be different based on our needs. If it is considered as the minimum value, most of the words will be selected as frequent and consequently a huge range of rules will be produced which most of them are not suitable. On the contrary, by considering a big constant, a small number of frequent words will be found that cannot satisfy our purpose. Based on our experiments, setting Min\_Document\_Repetition to 2 provides the most effective results.

* 1. producing subsets of frequent words and saving rules

As mentioned above, Table 6 shows the frequent words that could be more useful in predicting the class label of unseen documents. Therefore, if we find the rules that are produced from these words then we have completed the phase of pruning before producing the rules. As a consequence, the domain of rules is greatly reduced.

For example, considering our example in in Table 6; there are (2^3)-1=7 subsets in class label C1, 3 subsets in C2 and 15 subsets in C3 whereas without this work, (2^6)-1=63 subsets for each class could be produced. It should be noted that in our example, the word W2 that is frequent for each class label, is identified as a stop word and is not included in the generation of rules. Overall, frequent words with more than 75% in class labels are identified as ‘stop words’ and removed.

At this stage the Apriori algorithm is applied. First, every subset of the class labels is generated. If the produced subset (that shows frequent words of a class label) found frequent, the rule will be created. In order to determine a subset as frequent words, a binary ‘AND’ operation will be applied between it and the content of Table 2. If the result of this 'AND' operation is the same as the subset, it means that the words in the subset will be contained in an unseen document. This is applied to all training documents and finally if the number of documents that satisfy this condition is more than the threshold, the subset is accepted as a predictive rule and is saved for next stage.

The generated rule consists of two parts. The generated subset of the words is placed in the left hand side of the rule and the name of the class label placed in the right hand side. The created rule states that if an unknown document contains the words of the rule, it could be predicted that the class label of this document will be same as the rule.

Since generating all the subsets of a set is of exponential order (); subsets with one word are first generated and the process is repeated for subsets of two words and so on, until no rule is produced with a specific length (for example when no rule with the length of 5 words is produced then other subsets will not be generated). In this situation, the value of threshold will be decreased by one and then the rules will be produced again with the last length which was produced before (after decreasing threshold, the rules with the length of 5 will be produced again). The process of decreasing the threshold will continue until reaching a specific constant. The best values for these constants are specified by experiments which are provided in Table 8.

|  |
| --- |
|  |
| Figure 3. An example of finding a rule |

Figure 3 is an example of finding subsets and counting and checking frequency for an item. We have considered C1 which contains three frequent words. In order to produce suitable rules, first we should generate all the subsets of C1 and then Binary 'And' should be applied for every subset of C1 with every row of Table 2. As it is shown in Figure 3, the result of logical ‘AND’ operator for the specified subset is the same with three training documents. If threshold = 3, the selected subset is considered as frequent words and saved as a rule.

Now we have found the rules which can predict the class label of an unseen document. Before doing nothing, first we save these rules to a table of words. In order to save generated rules, a process similar to the previous steps is followed. In this case, all the used words in the rules will be identified and then a table of Boolean values is created so that its columns are words used in the rules and its rows are rule numbers. In every row, the cell of a word used in the rule then becomes 'True' and the others become 'False'. Thus a table is created so that in each row the constituent words of the rules are 'True'.

Among the rules created, there are some rules where the constituent words are the same but they predict different classes. For example rules like and . In these circumstances, similar rules should be deleted, because they are occasions of redundancy of created results and do not correctly predict the class labels.

* 1. Classifying unseen documents

In this step, a binary table is created so that its columns are key words (words that are used in rules) with the order of key words' list and its rows are the indices of pre-processed unknown documents. In the next stage, for each document if the key word is used in it, the cell of that word for the document becomes true. It means that we are going to find key words in unknown documents. After filling this table, we have found all the important features of the unknown documents. Now it is sufficient to binary 'AND' all rules (saved in binary format) with all rows of the new table. If the result of the binary 'AND' operation is equal to the binary value of the rule, it means that the words of the rule are identified as key words for the unknown document. So this rule can predict the class label of the document.

After finishing this step, a set of rules that can predict the class label of the document is collected. At the end, we will examine predicted class labels by counting generated rules for each document. The goal is to extract class labels with the maximum number of rules.

|  |
| --- |
|  |
| Figure 4. A sample of predicting unknown document |

Figure 4 is a sample of predicting the class label of an unknown document. In this sample every rule could consist of six words (because the length of each rule is 6) and only three words of these six words are used in the unknown document (because three bits have True value). In order to predict the class label of this document, an 'AND' operation should be done between the bit stream of the document and all the rules. As aforementioned, if the result of this operation is the same as the word-set of the rule, it shows that all the words that formed the rule exist in the unknown document so this rule could be used for prediction for the document. In Figure 4, it is clear that seven rules could be used for prediction and in this situation the class label that most of the rules refer to is selected.

In general, we may encounter the same number of votes for some classes which is natural. In the real world we cannot usually categorize a news article into specific class labels. For instance a number of articles are both political and Financial. In this situation due to structure of the algorithm where only one prediction should be proposed, one class label is selected randomly.

1. Experimental Results
   1. Data

As in any text mining approach, pre-processing is the first phase of the proposed algorithm. Tokenization, normalization and stemming are the main tasks of pre-processing phase which help to convert documents to structured format. The focus of our proposed algorithm is on the output of pre-processing phase. In addition, because of the binary nature of the approach, documents from any language can be used to evaluate the performance of the algorithm.

To demonstrate the performance of the algorithm, a dataset of Persian news articles, containing 565 documents, including five classes; social, financial, cultural, political and sport have been collected. This dataset contains almost equal numbers of documents in each class which are collected from several Persian news agencies [28][29][30][31]. The categories of the articles have been determined according to the classification of the news agency.

* 1. Cross Validation

A supervised learning algorithm needs some of data to be labelled as training data and some of them as test data [32]. These two datasets must be separate to prevent false results in evaluating the performances of the methods. Therefore multiple runs of the experiments are usually required with different datasets at each turn. One of the approaches splitting the dataset and running the experiment is N-fold cross validation. N-fold cross validation consists of splitting the dataset into N subsets of equal size. At each turn, one set is used for testing and the rest for training the system. In our case, 4-fold cross validation is used. At each turn, one fold have been used for training and learning and three for testing, in such way that every subset will be used once for testing purposes.

* 1. Evaluation metrics

We use precision, recall and F-score to measure the effectiveness of the proposed approach.

|  |  |  |
| --- | --- | --- |
| **Table 7**. Contingency table | | |
|  | Not Predicted as Feature | Predicted as Feature |
| Wrong Features | TN | FP |
| True Features | FN | TP |
|  |  |  |

The precision, recall and F-score are calculated based on Table 7 as:

(1)

(2)

(3)

To deal with multiple datasets (products), we adopt the macro and micro average (Yang 1999) to assess the overall performance. The macro- and micro-averaged precision and recall across the n datasets are defined as follows:

(4)

(5)

(6)

(7)

The macro average is calculated by simply taking the average obtained for each dataset, which assigns an equal weight for every dataset and product. Whereas the micro average assigns each dataset a relative weight on the basis of the number of extracted or manually extracted features for the dataset.

* 1. Comparative study

Raw documents are pre-processed into word vectors. For this purpose, a Persian pre-processing method [33] is applied to the dataset. In this step, redundant words, characters and symbols are removed and words with the same root are stemmed. Then the above algorithms are used to classify the documents. Applying stemming strongly reduced the dimension of the vectors, so the number of produced rules was reduced and subsequently time consumption of the algorithm was reduced.

In order to find the best results for BACA, different ranges for minimum Support and minimum confidence are examined. The results show that considering the lowest value for confidence (Conf >0) and a combination of different support ranges for different class labels gives the best answer. The reason of this result is due to the way that we have produced the rules. As mentioned before, first we find the effective words and then we produce the rules. Since Association Classification algorithms produce readable and understandable rules, they could be modified by domain experts. After reading the rules, we notice that by considering different ranges of supports for different class labels, the overall result is improved (Table 8). It should be noted that the thresholds which are used in our algorithm, are related to the words of the documents. Each document consists of a finite range of words. The number of words' repeated in the documents is important for our proposed algorithm which is not related to the number of documents that are classified.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table 8.** Comparison of accuracy of different classes in different range of supports | | | | | |
| Range of Supports | Financial | Social | Cultural | Political | Sport |
| 8-6 | 7 | 62 | 71 | 56 | 96 |
| 7-6 | 56 | 47 | 68 | 55 | 96 |
| 6-6 | 24 | 84 | 60 | 80 | 97 |
| 6-5 | 24 | 84 | 60 | 80 | 97 |
| 6-4 | 0 | 91 | 53 | 86 | 96 |
| Combination | 61 | 80 | 52 | 85 | 97 |
|  |  |  |  |  |  |

As mentioned in section 4, BACA consist of two parts - firstly feature generation and secondly the classification algorithm. In order to show better evaluation of the proposed algorithm, different parts of the algorithm have been compared separately with some known methods of performing each part. To begin with, we have compared generated features by proposed method (BACA) to those obtained by TF\_IDF [34] and Entropy [35] methods. Figure 5 shows the comparison of the accuracy measure of BACA, TF\_IDF and Entropy for five main classes, namely; cultural, financial, political, social and sport classes. It is clear that in each class, BACA presents better performance in terms of feature generation. BACA has shown significant improvement in generating features for political, sport and cultural documents which are 17.51%, 15.41% and 12.83% higher than Entropy respectively. Overall, TF\_IDF does not have good accuracy for all classes in comparison with BACA and Entropy.

|  |
| --- |
|  |
| **Figure 5.** Comparison of accuracy measure in different classes for Entropy, TFIDF and BACA |

To assess the overall performance of techniques, Figure 6 shows the macro and micro average for precision and recall respectively. As shown, the macro-averaged precision and recall of the Entropy are 60.84% and 53.08% respectively. Moreover, the macro-averaged precision and recall of the TF\_IDF are 40.56% and 29.19%. Whereas, the macro-averaged precision and recall of BACA are 78.64% and 57.53% respectively. Therefore, the effectiveness of BACA is better than that of Entropy and TF\_IDF, as presented in macro-averaged precision and recall. Specifically, macro-averaged precision obtained by BACA is 17.8% and 38.08% higher than those reached by Entropy and TF\_IDF. BACA reaches to a macro-averaged recall at 57.53% which improves the Entropy by 4.45% and TF\_IDF by 28.34%. Considering the micro average measures, we observe similar results to those we achieved by using macro average measures.

|  |
| --- |
|  |
| **Figure 6.** Comparison of Macro and Micro average for precision and recall for Entropy, TFIDF and BACA |

To evaluate the performance of the second part of the proposed algorithm, the algorithms of SVM, NB and KNN [5] have been used. In KNN algorithm, distance type is cosine and weighting type is LTC. In addition cross validation has been used to determine k=20 for the number of nearest neighbours.  For this purpose, after pre-processing and generating features, these algorithms are used for classification in the same condition with BACA. Figure 7 shows the results.

|  |
| --- |
| ` |
| **Figure 7**. Comparison of Accuracy measure between KNN, NB, SVM and BACA |

Figure 7 clearly shows the accuracy of BACA in two classes (Financial and Sport) is quite clear; in two other classes (Cultural and Political) the results are almost identical. However, in social class, BACA is weaker than other algorithm. One reason could be that the definition of social news is not very clear and it could cover various topics like social activities and various topics related to the society. Therefore, it covers a wide range of words which might not suitable for BACA classification. These different results in various classes could be owing to the different types of words used in documents of different classes.

|  |
| --- |
| ` |
| **Figure 8**. Comparison of Macro and Micro average for precision and recall KNN, NB, SVM and BACA |

To assess the overall performance of techniques, Figure 8 shows the macro and micro average for precision and recall respectively. From the figure, the macro-averaged precision and recall of the SVM are 50.9% and 23.02% respectively. Moreover, the macro-averaged precision and recall of the Naïve Bayes are 44.61% and 35.18%. Moreover, KNN has 46.7% and 32.06% for macro-averaged precision and recall. Whereas, the macro-averaged precision and recall of BACA are 67.91% and 45.03% respectively. Therefore, the effectiveness of BACA is better than that of SVM, NB and KNN as presented in macro-averaged precision and recall.

1. Conclusion

A new association classification algorithm, named BACA has been proposed that aims to classify documents. The proposed algorithm has many improved features against traditional and association classification methods which are used for document classification. It (1) produces understandable rules that can easily be interpreted by humans, (2) contains a feature selection part in order to decrease text dimension, (3) clusters important features based on class labels, (4) requires only one scan on training data, (5) saves features as a binary stream and uses binary operations in all the processes in order to reduce the memory space needed, (6) has the ability to predict more than one class label for each unseen document (Multi-label classification); (7) it is independent of language in classification.

Using association classification algorithms in text classification is a new method that has been used recently by some researchers. Our experiments on a dataset of Persian documents of five different classes, collected from diverse news agencies, show that BACA can perform well and has shown better accuracy in comparison with KNN, NB and SVM.

As future work, we propose to implement similar association rule based algorithms and compare their performance with ours. In addition investigating the multi-labelling feature of our classification algorithm could be of interest to a researcher in this domain.

Funding

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

References

[1] zu Eißen SM, Stein B. Genre Classification of Web Pages, [Advances in Artificial Intelligence](http://link.springer.com/book/10.1007/b100351) [Lecture Notes in Computer Science](http://link.springer.com/bookseries/558) Volume 3238, 2004, pp. 256-269.

[2] Pang B and Lee [L.](http://scholar.google.com/citations?user=oJESe-cAAAAJ&hl=en&oi=sra)  Opinion Mining and Sentiment Analysis, Foundations and trends in information retrieval, Volume 2 Issue 1-2, January 2008, pp. 1-135.

[3] [Dumais](http://scholar.google.com/citations?user=x8dED5cAAAAJ&hl=en&oi=sra) S and [Chen](http://scholar.google.com/citations?user=1Aa3qxIAAAAJ&hl=en&oi=sra) H. Hierarchical classification of Web content, Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, 2000, pp. 256-263.

[4] Androutsopoulos I, Koutsias J, Chandrinos KV, Paliouras G, Spyropoulos CD. An evaluation of Naive Bayesian anti-spam filtering, 11th European Conference on Machine Learning, Barcelona, Spain, 2000, pp. 9-17.

[5] Wu x, Kumar V, Quinlan J.R, Ghosh J, Yang Q, Motoda H, McLachlan G.J, NG A, Liu B, Yu P.S, Zhou Z, Steinbach M, Hand D.J, Steinberg D, Top 10 algorithm in data mining, [Knowledge and Information Systems](http://link.springer.com/journal/10115) 2008, Vol. 14, [Issue. 1](http://link.springer.com/journal/10115/14/1/page/1), pp. 1-37 .

[6] Li W, Han J, Pei J. CMAR: Accurate and efficient classification based on multiple-class association rule, In ICDM'OI, San Jose, CA, Nov. 2001, pp. 369-376.

[7] Lie B, Hsu W, Ma Y. Integrating classification and association rule mining, In KDD'98, New York, NY, Aug. 1998, pp. 80-86.

[8] Agrawal R, Srikant R. Fast algorithms for mining association rule, Proceedings of the 20th International Conference on Very Large Data Bases, 1994, pp. 487- 499.

[9] Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation, In SIGMOD'00, Dallas, TX, May. 2000, pp. 1-12.

[10] Yoon Y, Lee G.G. Text categorization based on boosting association rules, In Proceedings of ICSC 2008, second IEEE international conference on semantic computing, pp.136–143.

[11] Yin X, Han J. CPAR: Classification based on predictive association rule, In SDM 2003, San Francisco, CA, May. 2003.

[12] Tang ZH, Liao Q. A New Class Based Associative Classification Algorithm, IAENG International Journal of Applied Mathematics, 2007, 36:2, IJAM\_36\_2\_3.

[13] Antonie M. L, Zaiane O. R. Text Document Categorization by Term Association,  [ICDM 2003, Proceedings 2002 IEEE International Conference on](http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8435) Data Mining, pp. 19-26.

[14] Wang J, Karypis G. Harmony: Efficiently mining the best rules for classification, In SDM, 2005.

[15] Thabtah F, Cowling P, and Peng Y, MCAR: multi-class classification based on association rule, Computer Systems and Applications, The 3rd ACS/IEEE International Conference on, 2005, pp. 33.

[16] Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases. In Proc. of the ACM SIGMOD Conference on Management of Data, Washington, D.C., May. 1993.

[17] Quinlan J. R, Cameron-Jones R. M. FOIL: A midterm report, In Proc. European Conf. Machine Learning, Vienna, Austria, 1993, pp. 3-20.

[18] Yoon Y, Lee G.G. Two scalable algorithms for associative text classification, Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH), San 31, Hyoja-Dong, Pohang 790-784, Republic of Korea, [Information Processing & Management](http://www.sciencedirect.com/science/journal/03064573),[Volume 49, Issue 2](http://www.sciencedirect.com/science/journal/03064573/49/2), March 2013, pp. 484–496.

[19] Blake C.L. and Merz, C.J, UCI Repository of machine learning databases, UCI Repository of Machine Learning Databases, 1998, URL http://archive.ics.uci.edu/ml/

[20] Hammoud S, A Thesis submitted for the Degree of Doctor of Philosophy, Brunel University, May 2011

[21] Agrawal R, Srikant R. Fast algorithms for mining association rules, In Proc. 20th Int. Conf. Very Large Data Bases, VLDB 1215, 1994, pp. 487-499.

[22] Han J, Pei J, Yin Y, and Mao R, Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach, Data Mining and Knowledge Discovery, vol. 8, 2004, pp. 53-87.

[23] Meretakis D, Fragoudis D, Lu H, Likothanassis S. Scalable association-based text classification, In Proceedings of 10th International Conference on Information and Knowledge Management, ACM Press, McLean, New York, US, 2000, pp. 373–374.

[24] Wong A. K. C, Wang Y. High-order pattern discovery from discrete-valued data, IEEE Transactions on Knowledge and Data Engineering, 1997, 9(6): pp. 877–893.

[25] Liu B, Hsu W, Ma Y. Integrating classification and association rule mining, In Knowledge Discovery and Data Mining, 1998, pp. 80–86.

[26] Thanh Nhan L.T. An Apriori-Based Frequent Item sets Mining Using Bit Streams, Department of Computer Science yunghee University, Global Campus Yongin, Gyeonggi, Korea, 2010, pp. 446-701.

[27] Antonie M, Zai"ane O.R, Coman A. Associative Classifiers for Medical Images, Lecture Notes in Artificial Intelligence 2797, Mining Multimedia and Complex Data, Springer-Verlag, 2003, pp. 68-83.

[28] Mehr News Agency, http://www.mehrnews.com, Accessed on March, 2012.

[29] Isna News Agency, <http://www.isna.ir>, Accessed on March, 2012.

[30] Irna News Agency, http://www.irna.ir, Accessed on March, 2012.

[31] Fars News Agency, http://www.farsnews.com, Accessed on March, 2012

[32] Hu M. and Liu B, Mining and Summarizing Customer Reviews, In Proc. ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining, 2004, pp. 168-177.

[33] Sheydaei N, Saraee M.H, Shahgholian A, An efficient and Robust Preprocessing technique for Persian Text (In Persian), ICEE 2013

[34] Wu H.C, Luk R.W.P, Wong K.F, Kwok K.L. Interpreting TF-IDF Term Weights as Making Relevance Decisions, ACM Transactions on Information Systems, 2008, Vol. 26, No. 3, Article 13.

[35] Okafor A, [Pardalos](http://scholar.google.com/citations?user=7rBzAKsAAAAJ&hl=en&oi=sra) P, Ragle M, Data mining via entropy and graph clustering, [Data Mining in Biomedicine](http://link.springer.com/book/10.1007/978-0-387-69319-4) [Springer Optimization and Its Applications](http://link.springer.com/bookseries/7393), 2007, Vol. 7, pp. 117-131.