



Polychotomiser for Case-based Reasoning beyond the Traditional Bayesian Classification Approach

Dino Isa

Faculty of Engineering and Computer Science
University of Nottingham, Malaysia Campus, Semenyih, Malaysia.
Tel: 603-89248116 E-mail: dino.isa@nottingham.edu.my

Lam Hong Lee

Faculty of Engineering and Computer Science
University of Nottingham, Malaysia Campus, Semenyih, Malaysia.
Tel: 603 89248141 E-mail: kcx4lhl@nottingham.edu.my

V.P. Kallimani

Faculty of Engineering and Computer Science
University of Nottingham, Malaysia Campus, Semenyih, Malaysia.
Tel: 603 89248141 E-mail: VP.Kallimani@nottingham.edu.my

R. Prasad

Faculty of Engineering and Computer Science
University of Nottingham, Malaysia Campus, Semenyih, Malaysia
Tel: 603-89248116 E-mail: Rajprasad.Rajkumar@nottingham.edu.my

Abstract

This work implements an enhanced Bayesian classifier with better performance as compared to the ordinary naïve Bayes classifier when used with domains and datasets of varying characteristics. Text classification is an active and on-going research field of Artificial Intelligence (AI). Text classification is defined as the task of learning methods for categorising collections of electronic text documents into their annotated classes, based on its contents. An increasing number of statistical approaches have been developed for text classification, including k-nearest neighbor classification, naïve Bayes classification, decision tree, rules induction, and the algorithm implementing the structural risk minimisation theory called the support vector machine. Among the approaches used in these applications, naïve Bayes classifiers have been widely used because of its simplicity. However this generative method has been reported to be less accurate than the discriminative methods such as SVM. Some researches have proven that the naïve Bayes classifier performs surprisingly well in many other domains with certain specialised characteristics. The main aim of this work is to quantify the weakness of traditional naïve Bayes classification and introduce an enhance Bayesian classification approach with additional innovative techniques to perform better than the traditional naïve Bayes classifier. Our research goal is to develop an enhanced Bayesian probabilistic classifier by introducing different tournament structures ranking algorithms along with a high relevance keywords extraction facility and an accurately calculated weighting factors facility. These were done to improve the performance of the classification tasks for specific datasets with different characteristics. Other researches have used general datasets, such as Reuters-21578 and 20_newsgroups to validate the performance of their classifiers. Our approach is easily adapted to datasets with different characteristics in terms of the degree of similarity between classes, multi-categorised documents, and different dataset organisations. As previously mentioned we introduce several techniques such as tournament structures ranking algorithms, higher relevance keyword extraction, and automatically computed document dependent (ACDD) weighting factors. Each technique has unique response while been implemented in datasets with different characteristics but has shown to give outstanding performance in most cases. We have successfully optimised our techniques for individual datasets with different characteristics based on our experimental results.

Keywords: Text Classification, Bayes Theorem, Bayesian Filtering, Probability, Case-Based Reasoning

1. Introduction

Document classification is defined as the task of learning methods for categorising collections of electronic documents into their annotated classes, based on its contents. For several decades now, document classification in the form of text classification systems have been widely implemented in numerous applications such as spam filtering. (Sahami et.al., 1998; Cunningham et. al., 2003; Delany, Cunningham & Coyle, 2005), e-mails categorising (Kamens, 2005), directories maintenance, and ontology mapping (Su, 2002), contributed by the extensive and active researches. An increasing number of statistical approaches have been developed to document classification, including k-nearest-neighbor classification, naïve Bayes classification, support vector machines, decision tree induction, rule induction, maximum entropy, artificial neural network, etc.

Each of the document classification schemes has their own properties. The decision tree induction algorithm and rule induction algorithm are simple to understand and interpret after a brief explanation. However, these algorithms do not work well when the number of distinguishing features is large (Quinlan, 1993). k-nearest neighbor algorithm is easy to implement and shows its effectiveness in a variety of problem domains (Han, Karypis & Kumar, 1999). A major drawback of the k-NN algorithm is computationally intensive, especially when the size of the training set grows (Han, Karypis & Kumar, 1999). Support vector machine can be used as a discriminative document classification method which has been shown to be more accurate in classification tasks (Joachims, 1998; Chakrabakti, Roy & Soundalgekar, 2003). The high accuracy of SVM is due to the implementation of structural risk minimisation which entails finding a hyper-plane which guarantees the low error plus an ability to learn which is independent of the dimensionality of the feature space (Joachims, 1998). However, the usage of SVMs is not popular in many real world applications due to its convoluted training and categorising algorithms (Chakrabakti, Roy & Soundalgekar, 2003).

Among these approaches, naïve Bayes text classification has been widely used because of its simplicity in both training and classifying stage although this generative method has been reported less accurate than discriminative methods such as SVM (Joachims, 1998; Chakrabakti, Roy & Soundalgekar, 2003). However, some researches have proven that naïve Bayes classification approach provides intuitive simple text generation models and performs surprisingly well in many other domains, under some specific conditions (McCallum & Nigam, 2003). A naïve Bayes classifier is a simple probabilistic classifier based on Bayes' Theorem with strong independence assumptions but this assumption severely limits its applicability. Depending on the precise nature of the probability model, naïve Bayes classifiers can be trained very efficiently and requires a relatively small amount of training data to estimate the parameters necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

Bayesian Filtering is a probabilistic inference approach which is typically implemented in mail repositories to remove spam e-mails (Sahami et.al., 1998; Delany, Cunningham & Coyle, 2005; Cunningham et. al., 2003; Delany et. al., 2004). Bayesian filtering is a highly effective approach in the task of classifying data with a very low number of false positives. It overcomes the obstacles faced by more static technologies. The traditional Bayesian Filtering approach is implemented as dichotomiser, a two-category classifier, which classifies e-mails into spam and legitimate, by calculating the overall probability of the text body of the document. However, our research has emphasised in developing an extended version of Bayesian classifier which is able to handle multiple categories classification tasks, and guarantee minimum error rate classifications. Our proposed Bayesian probabilistic classifier has been implemented in conjunction with a self-organising map (Hartley et. al., 2006) in a case-based reasoning system which contribute to an efficient and time saving case retrieval process. Figure 1 illustrates the block diagram of our proposed case-based reasoning system's structure, which is enhanced with the extended Bayesian classifier in this work.

In the context of classification, Bayes theorem emphasises that the probability that a particular document is annotated to a particular category, given that the document contains certain words in it, is equal to the probability of finding those certain words in that particular category, times the probability that any document is annotated to that category, divided by the probability of finding those words in any document.

$$\Pr(\text{Category} | \text{Word}) = \frac{\Pr(\text{Word} | \text{Category}) \cdot \Pr(\text{Category})}{\Pr(\text{Word})}$$

Each kind of text documents contains words which are given probabilities based on its number of occurrence within that particular kind of documents. Bayesian filtering is predicated on the idea that spam e-mails can be filtered out based on the probability that certain words will correctly identify a piece of e-mail as spam while other words will correctly identify a piece of e-mail as legitimate. At the basic level, a Bayesian filter examines a set of e-mails that have been categorised to be spam and legitimate, and compares the content in both categories in order to build a database of words and their occurrence. The list of words occurrence is used to identify or predict future e-mails as spam or legitimate, according to the probability of words from the whole body of the text message of an e-mail.

Many researches have been carried out to implement Bayesian probability theories in case-based reasoning approach for case retrieval purposes. Bayesian network is one of the popular mechanism which many research groups have

investigated and developed new approaches for case matching algorithm in case-based reasoning (Myllymaki & Tirri, 1993; Myllymaki & Tirri, 1994; Kontkanen et. al., 1997; Kontkanen et. al., 1998). In this paper we emphasis on the Bayesian filtering approach to be implemented as an extensive version of classifier and acts as a part of our proposed case-based reasoning's case retrieval algorithm.

In order to perform the task as a probabilistic classifier at the front end of case retrieval algorithm, the traditional Bayesian filtering approach needs to be extended to handle different types of multiple categorised data efficiently. The solutions database in a knowledge repository can be divided into multiple categories according to their similarity of properties, attributes and features. This classification leads to the efficient and time-saving case retrieval since the solution cases are well-organised by the self-organising map. A solutions database can be classified into linear multi-dimensional organisation, or hierarchical organisation, or even the integration of the above methods. Therefore, our classifier is designed to be highly adaptable to these databases with different organisations.

The proposed Bayesian classifier in our research is equipped with some specialised algorithms to ensure the high performance in handling different types of knowledge domain with unique characteristics and guarantee a low error rate classification. An extendable rank classification algorithm and a series of tournament structures algorithms have been designed and implemented in our proposed system. Besides, the polychotomiser, which represents the multiple categories classifier, is equipped with some extra techniques to guarantee the high accuracy of classification tasks.

2. The Classification Algorithms

2.1 Rank Classification Algorithm

The rank classification algorithm slightly modifies the traditional Bayesian filtering probability calculation in order to support multi-category dataset. The probability value for a document A to be annotated to a category C is computed as $\Pr(C|A)$. As an assumption that we have a category list as Cat1, Cat2, Cat3, Cat4, Cat5,, CatN, thus, each document has N associated probability values, where document X will have $\Pr(\text{Cat1}|X)$, $\Pr(\text{Cat2}|X)$, $\Pr(\text{Cat3}|X)$, $\Pr(\text{Cat4}|X)$, $\Pr(\text{Cat5}|X)$,, $\Pr(\text{CatN}|X)$. The probability value of an input document to be annotated to a particular category is calculated by considering all the categories together in a single round. The rank classification algorithm is then sorts all the probability values for each category and selects the highest score as the most likely category to be annotated.

A multi-category Bayesian classifier needs to fulfill some conditions to guarantee the high accuracy of classification. One of the conditions is that the classifier needs to be trained on an approximately equal number of training documents for each category due to the imbalance of sizes between categories may cause great risk in misclassifications. The greater the imbalance, the worse problems occur. This is due to the prior probability $\Pr(\text{Category})$, is computed as $1/\text{the total number of category}$. This equation is made based on the assumption that training set is perfectly balanced with the same number of training documents for each category. Furthermore, even the number of training documents for each category is balanced, the problems still occur since the size of each training documents is not approximately equal to each other. Therefore, as the solution for the problem of imbalance of training dataset, the prior probability for each category, $\Pr(\text{Category})$, is transformed from the ordinary $1/\text{the total number of category}$, to the equation:

$$\begin{aligned} \Pr(\text{Category}) &= \frac{\text{Total_of_Words_in_Category}}{\text{Total_of_Words_in_Training_Dataset}} \\ &= \frac{\text{Size_of_Category}}{\text{Size_of_Training_Dataset}} \end{aligned}$$

2.2 Round Robin Tournament Algorithm

Tournament structures are possible to be implemented in the classification algorithms to handle multi-category classification. Previous researches proved that the round robin tournament algorithm can performs the spam e-mails filtering beyond the ordinary binary classification (Kamens, 2005). In a round robin tournament algorithm, the calculation of the Bayesian probability values is performed only between two categories. It is a looping binary classification algorithm, and each competitor plays against all the others an equal number of times, typically once. The round robin tournament algorithm contributes to a relatively simple and complete competition among all the categories, and the process iterates until every category has compete against all the others.

The structure of the round robin tournament algorithm in our proposed system is a "Host and Guest" concept. Firstly, all the categories are randomly sorted. The first category will be the host for the initial round and plays against all the others sequentially which ranked below it as guests. At the second round, the second category will become the host for the second round and those categories which are ranked below it will compete against the host. The process iterates until all categories have played against the others an equal number of time. There are some methods available to determine the final winner after the iterative calculation processes complete. One of the methods is the winning category of each match is awarded with a score, typically 1, and the loser is not awarded with any score, or in the other words, score 0 is awarded. The scores from every match of a particular category are added together after the competition until the calculation has completed. The category with the highest score will be the overall winner, which represents the right category for the input document to be annotated.

There is a situation where dilemma occurs in determining the right category of an input document when more than one category which have the same highest final score. This situation can be avoided by awarding the two competing categories of each match with the score which is equal to their probability value computed from the binary classification. With this method, the final highest score for every category is rarely to be the same.

As the result, the round robin tournament has an improved ability to distinguish similar categories, since it is a looping binary classification algorithm. The binary classification algorithm can easily differentiate between two similar categories that both have great differ from the others. It is smart enough to isolate two categories temporarily and perform the probability values calculation without considering other parties. However, the iterative binary classification process consumes a relatively long time compare with other algorithms.

2.3 Single Elimination Tournament Algorithm

By comparing with other algorithms, the single elimination tournament structure has some restrictions. Firstly, most often the number of competitors is fixed as a power of two. Somehow, in the situation that the number of participants is not a power of two, typically the highest-rated competitors from the previous accomplishment will be advanced to the second round without joining any match in the first round. Besides, seeding is recommended as a pre-process to prevent highest-rated competitors being scheduled to face each other in the early stages of the competition. The seeds ranking process can be executed by using the rank classification algorithm or the round robin tournament algorithm. Therefore, the single-elimination algorithm is more suitable to be implemented at the back-end of an integrated algorithm.

As similar to the round robin tournament algorithm, single-elimination algorithm also performs the probability values calculation in the form of binary competition for every match. In the first round, we play the best competitor against the worst, and the second best against the second worst, and so on for the rests. Brackets are set up, so that the top two seeds could not possibly meet until the final round, none of the top four can meet before the semifinals, and so on. This concept is applicable in the following rounds until the overall winner is representing by the winner of the final round.

In contrast to round robin tournament structure, as rounds progress, the successive rounds of the single elimination tournament structure halve the number of remaining competitors by progressing the winners from the previous round to the next round and eliminating the losers. Single-elimination tournament algorithm is suitable to be implemented in the domains which have a large number of categories. Somehow, since this algorithm is also a binary classification based algorithm, it has great ability in handling the classification tasks which involve categories with high degree of similarities.

2.4 Swiss System Tournament Algorithm

In our proposed system, the Swiss system tournament algorithm can be implemented independently or at the back-end of a hybrid algorithm. The initial seeding of a Swiss system tournament is not a compulsory as the single-elimination tournament algorithm, but it is recommended. The competing categories are then divided into two parts, the top half which is paired with the bottom half. As an example, if there are eight categories in the classifier, first category is paired with fifth category; the second is paired with the sixth, the third is paired with the seventh, and so on.

After the first round of the competition, the winners from the first round will plays against the winners, and the losers will plays against the losers. As similar to the round robin tournament, the winning category of each match is awarded with a score, typically 1, and the loser is not awarded with any score. In further rounds, each competitor will be pitted against another competitor who has the same score. Modifications are then made to prevent competitors from meeting each other twice.

In contrast to round robin tournament, the Swiss system tournament algorithm can determine the top ranked and bottom ranked competitors with fewer rounds, although the middle rankings are unreliable. By the way, we only have interest on the final overall winner. Therefore, the Swiss system is applicable in the classification tasks of our system with large number of categories, and similar to other binary classification based algorithms, it is suitable to be implemented in the classification tasks which the domain contains the categories with high degree of similarities.

However, the number of competing categories has becomes the biggest obstruction for the Swiss system tournament algorithm. As similar to the round robin tournament, the Swiss system tournament algorithm has the potential in facing a dilemma in determining the annotated category of an input document, or the final winner. It may happen that two or more categories have the same highest and perfect score, won all the games but never faced each other. Therefore, the ordinary algorithm needs to be slightly modified in terms of the number of rounds played. To determine a clear overall winner, we have applied the same concept with single-elimination tournament algorithm in terms of number of rounds that is the base 2 logarithm of the number of competitors rounded up.

3. The Low Error Rate Classification Techniques

3.1 Keywords Counting Methods- Multinomial and Multivariate Technique

In our proposed system, as similar to other approaches, the classifier must be trained in advance so that it can build up all the probability values for every recognised keyword to be annotated to every category. The knowledge engineers or the domain experts must manually organise the training dataset which contains a reasonable number of training files for each

category, and then the classifier will accordingly adjust the words' probabilities in every category of the database. In the training phase, we first need to analyse the training files by extracting all the words to generate a list of words occurrence frequency. Our prototype system, which is developed by using JAVA, analyses the plain text documents from the training set to generate a list of words occurrence frequency for each category, and the list of words occurrence frequency for every category is stored independently by using a data structure, as a TreeMap of String word -> Integer frequency. The occurrence of a particular word in the list of words occurrence for a particular category is given by the value regards to the total occurrence of that particular word in all the training files from that particular category, so called multinomial method.

The multinomial keywords counting method is suitable to be implemented for the training dataset which has the approximately balance sizes of categories. If the training dataset is under the imbalance condition in terms of the size of each category, problems may occur since that each indicative probability score is very low, and the problems may become serious as the greater imbalance of training data. This problem can be solved by collecting training files which has the approximately same size.

Multivariate keywords counting method is an alternative method which has better performance as compared to the multinomial method under the condition of imbalance training dataset. The multivariate method calculates the words occurrence frequency for every category based on the number of training files which contain the particular word in a particular category. For example, if a particular word is found in a number of training files which have been categorised under a particular category, the occurrence of that particular word in the list of words occurrence for the particular category is given by the value regards to the total number of training files which contain the particular word in that particular category.

However, the multivariate method is not suitable to be implemented in the cases which the categories are very similar to each other. The results from our experiments in handling dataset with high degree of similarity show that multinomial method performs better than multivariate method. This can be concluded as the training files of similar categories contain most of the same keywords with each other. The frequency of usage of a particular keyword in different categories is the key factor to differentiate these similar categories. Therefore, the classifiers which handle categories with high degree of similarities are preferably to be implemented with the multinomial keywords counting method.

3.2 High Relevance Keywords Extraction Facility

The ordinary Bayesian filtering approach takes the whole message into account to identify whether an e-mail is spam or legitimate (Sahami et. al., 1998). Improving on this classification method, we have proposed, for our text classifier, a technique which identifies the high relevance keywords during the fly of the classification process, not from a pre-defined keywords dictionary. When the system has analysed the input document or input query the input text is stored into a class Scanner instead of String since Scanner can parse primitive types and strings using regular expressions. During the calculation of the probability values, each individual word from Scanner is extracted to calculate for the probability to be annotated to each of the categories. However, only words which have relatively high probability value, or "Important" to be annotated to a particular category compare with the others will be taken into account. The degree of "Importance" of keywords can be adjusted by setting a threshold. Based on our experiments, the greater the threshold, the better the classifier performs. However, the performance decreases when the threshold reaches a certain limit, which is the saturation point of the degree of difference. The saturation point of the degree of difference is domain-dependent.

3.3 ACDD Weighting Factors Facility

The research has found that in certain situations, certain categories have high number of misclassified documents. In other words, a large number of documents are likely to be misclassified in a wrong category. This problem may be caused by reasons such as the imbalance training datasets and the improper organisation of the training database resulting in the mis-training of the system.

This work has proposed a solution for the problem mentioned above, which is the implementation of the automatically computed document dependent (ACDD) weighting factors to the probability values of the input data. Different values of weighting factor are applicable to different categories in the classification tasks. Categories which have high intensity of correctly classified data will be awarded with a relatively small value of weighting factor. On the other hand, relatively large value of weighting factors will be awarded to categories which their documents are always been misclassified as others. For the ACDD weighting factors calculation session, a process which is similar to the calculation of probability values of input documents, is executed by getting another set of data, namely ACDD weighting factors retrieval dataset as the input. Therefore, to implement the ACDD weighting factors method, an initial training dataset, and an ACDD weighting factors retrieval dataset are needed for the pre-process for the classifier before it can start its classification tasks.

The ACDD weighting factors retrieval dataset can be organised by the domain experts, or extracted from the training dataset. The system is initially trained with the training dataset before performs the weighting factors computation. The ACDD weighting factors is calculated by loading the ACDD weighting factors retrieval dataset as the input data and the

results are recorded for the purpose of generating the weighting factors for every category. Weighting factor of a particular category is computed based on the total number of documents from the ACDD weighting factors retrieval dataset which are annotated to it. The more documents are annotated to the category, the smaller the value of its weighting factor.

After the ACDD weighting factors computation, every category has been awarded by a unique weighting factor. During the classification process of the input data, the probability values of the document are calculated based on the same training dataset and the same algorithm and methods. The probability values of a document to be annotated to each category are added with their own unique weighting factor before the system determines the annotated category of the input document.

Results from our experiments proved that the ACDD weighting factors implementation successfully reduces the misclassification rates. However, among the disadvantages of this approach is that it consumes a relatively long processing time compare with other approaches, and required a complex algorithm to compute the weighting factors automatically. The system needs to process the ACDD weighting factor retrieval dataset after the training session to compute the weighting factors for every category. Besides, a complex and iterative algorithm is needed to determine the formula and the multiplier for the computation of the weighting factors for every category, which need to be tested iteratively until a set of optimum ACDD weighting factors for every category is generated.

4. The Experimental Results

The objective of these evaluations is to determine whether our proposed approaches and methods resulted in better classification accuracy and performance compare with the ordinary version, which will greatly contribute as the front-end classifier for case retrieval stage of case-based reasoning system.

The evaluations of our prototype system are executed by applying different kinds of knowledge domain, different probability values calculation algorithms and different low error rate classification methods. As mentioned in the sections above, every classification algorithm and low error rate classification method has different performance, depending on the characteristics of the knowledge domain. Therefore, to have an optimum Bayesian classifier, the flexibility of the classifier is needed by selecting right techniques to the right knowledge domain.

4.1 Experiment 1: Dataset with Low Degree Similarity and Imbalanced Category.

The dataset of variants of vehicle is tested by our prototype system for the evaluation of classification performance in handling the case with categories which have low degree of similarity. Our selected dataset contains four categories of vehicles: Aircrafts, Boats, Cars, and Trains. All the four categories are easily to be differentiated and every category has a large number of their own unique keywords. We have collected 90 documents for each category, with the total of 360 documents in the entire dataset. 30 documents from each category are extracted randomly to build the training dataset for the classifier. The rest of 60 documents for each category are remained as the testing dataset to test the classifier.

Initially, we have performed the experiment by implementing different classification algorithms: the rank classification, round robin tournament, single-elimination tournament and Swiss system tournament. The goal of this experiment is to compare the performance of these algorithms. Figure 2 illustrates the comparison chart of the performances of each algorithm in this experiment.

The results illustrated in Figure 2 show that the rank classification algorithm performs the best among the other classification algorithms, with the classification rate of 88.89%. The rank algorithm is a direct classification approach which performs the probability values calculations in one round and consumes the shortest time as compared to the others.

The round robin tournament algorithm performs the best among the tournament structure based algorithms since it contributes to a relatively simple and complete competition among the categories. The classification rate of round robin tournament algorithm is 79.50%. However, the complete competition among the categories is due to the round robin tournament algorithm executes the binary classification process iteratively. Therefore, relatively long time consumption is required by this algorithm.

The single-elimination tournament algorithm has the similar performance with the Swiss system tournament algorithm, where the classification rate is 76.82% for single-elimination and 76.44% for Swiss system. However, both of these algorithms need a pre-process for the initial seeds ranking, which is a compulsory for the single-elimination tournament algorithm but is an optional for the Swiss system. Hence, the results show that the integrated versions are not performing as good as the independent versions of classification algorithm in terms of performance and time consumption.

Another experiment is also been carried out to justify the contribution of the low error rate classification methods. By applying these methods to the classifier which implements the rank classification as the basic algorithm, a comparison of the performance of these low error rate classification methods is illustrated as Figure3.

The results illustrated in Figure 3 above show that the keywords occurrence counting based on the multivariate method performs better with the classification rate of 88.89%, while the keywords occurrence counting based on the multinomial

method only contributes to a classification rate of 64.94%. The reason for the relatively high misclassification rate of multinomial method is due to the imbalance of the size of categories in the training dataset. Problems may occur due to the high frequency of occurrence for certain keywords in big size categories which may lead to the classifier misclassified most of the input documents to them.

Besides our prediction for the better performance of the implementation of High Relevance Keywords Extraction method in overall, the HRKE also performs better while threshold for the "Importance" of keywords increase. The High Relevance Keywords Extraction method contributes to 89.08% classification rate with the threshold is set at 40%, and 95.21% with the threshold is set at 90%, and 97.92% with the threshold is set at 95%. This method culls out all the common and disregarded words for every category, and also the words which have the similar effect to the classification task. Therefore, the confusion during the classification task for the input documents is reduced and this will leads to a higher classification rate.

4.2 Experiment 2: Dataset with High Degree Similarity and Balanced Categories

Another dataset which has been tested by our prototype system is the dataset of mathematics topics, which has a high degree of categories' similarity. Due to these topics are the sub-topics of mathematics subject, the common mathematical terms are widely used by the documents from these topics. There are only a few specific and unique terms to differentiate each topic, therefore the degree of similarities for the categories from this dataset is relatively high compare with the Vehicles dataset. 10 files for each category have been organised as the training database, while the testing dataset contains 240 files for testing purposes.

As similar to the previous experiment, we have tested our classifier with the proposed classification algorithms. We have executed each of these under the basic condition by not implementing any low error rate classification methods for the goal of pure comparison of the performance between the classification algorithms. The chart illustrated in Figure 4 shows the comparison of the performance of each algorithm in this experiment.

The pattern of this comparison is different from Experiment 1. The rank classification algorithm contributes to a lower classification rate of 80.42% since it calculates the probability values of every category for an input data in one round and takes all categories into account. This will lead to a great confusion to the classification tasks since all the categories are similar to each other in terms of words usage. In such a situation, binary classification based algorithms are able to overcome this problem. Therefore, all the tournament structure based algorithms perform better than the rank classification algorithm, with the same classification rate of 81.25% which is slightly higher than the rank classification algorithm. Although all the three tournament structure based algorithms in this experiment have the same classification rates co-incidentally, the patterns of classification for each of them are different.

The following test is to figure out the comparison of performance of the low error rate classification methods. The rank classification algorithm has been chosen as the basic classification algorithm. The chart in Figure 5 illustrates the comparison of performance of every low error rate classification methods for this high degree of similarity and balanced dataset.

From the comparison chart illustrated in Figure 5, we concluded that for the classifier to handle dataset with high degree of categories' similarity, the multinomial keywords counting method performs better than the multivariate method, which is 80.42% compare to 77.92%. This is due to majority of the documents in every category contain the same keywords since the degree of similarity is high. To differentiate these highly similar categories, the keywords counting based on the number of keywords occurrence in all the training files of the categories is required since different topics may have different frequency of usage for a particular keyword.

The High Relevance Keywords Extraction method does not act the same pattern with Experiment 1. In this experiment, every word in the documents may bring great effect to the classification tasks. The increment of the threshold leads to the less words to be considered. However, the common and disregarded words can be culled out with a low threshold, which can slightly increase the classification rate of the classifier. When the value of the threshold increases, the important keywords for the classification task are not taken into account. Therefore, the value of threshold in such a situation should not over the saturation point of the degree of difference.

The most significant method which we have discovered in our research is the ACDD weighting factors implementation. Weighting factor is a variable component used in calculations that allows for a margin of error above the minimum error on a measurement, or produces a desired result. This method contributes to 88.90% of classification rate, compares to the others which are lower than it. This is due to the weighting factors are able to reduce the intensity of misclassified data for some "popular" categories, and contributes to the increment of the classification accuracy. The highest classification rate from this experiment by implementing the ACDD weighting factors is 88.90%. Experiment 1 scored the highest classification rate at 97.92% by implementing the HRKE facility, which can be considered as an almost perfect classification task. By referring back to the number of training files for both experiments, Experiment 2 takes the training dataset which contains 10 training files for each category while Experiment 1 has 30 training files for each category in the

training dataset. From here we can conclude that this system is acting like other Artificial Intelligence application, “The More It Learns, The Smarter It Works”.

5. Conclusion and Future Works

The case retrieval approach for a case-based reasoning system through the implementation of the Bayesian Filtering technique at the front-end, in conjunction with a self-organising map at the back-end, has been proposed and developed by our research group. This approach takes the advantages of Bayesian probability theorem which may greatly enhance the performance of ordinary case retrieval algorithms. Besides, we introduce a series of classification algorithms and low error rate classification methods to develop an extensive version of Bayesian probability classifier. The results from our experiments show that the Bayesian probability classifier is required to be flexible and optimised for different datasets. This can be done by implementing different algorithms and methods to match the requirements of different characteristics of variety of dataset. We have established the optimal requirements for the classification tasks for specific dataset with different characteristics. The experimental results show that the properties of all the proposed classification algorithms and also the techniques used for the low error rate classification can be optimised for any domain of different characteristics. In the future, our research group is emphasising on enhancing the ability and performance of our existing prototype by introducing some facilities, such as natural language processing approach to handle sentences classification, not restricted to individual keywords classification. Our group hopes to extend the advantages of Bayesian probability theorem, not restricted to categorise data, but towards more AI applications such as in sensor monitoring (Isa et. al., 2007).

References

- Aamodt, A., Plasa, E. (1994). Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AICom - Artificial Intelligence Communications, IOS Press, Vol. 7: 1*, pp. 39-59.
- Agre, G., Koprinska, I. (1996). Case-Based Refinement of Knowledge-Based Neural Network. *In Proceedings of the International Conference, Intelligent Systems: A Semiotic Perspective*, October 20-23, Gaithersburg, MD, USA.
- Bergmann, R., Richter, M. M. Schmitt, S. Stahl, A. Vollrath, I. (2001). Utility-Oriented Matching: A New Research Direction for Case-Based Reasoning. *In Proceeding of the 9th German Workshop on Case-Based Reasoning, GWCBR'01, Baden-Baden, Germany*.
- Block, S., Medin, D., & Osherson, D. (2002). Probability from Similarity. Northwestern University; Rice University.
- Burkhard, H. D. Richter, M. M. On the Notion of Similarity in Case-Based Reasoning and Fussy Theory. Soft computing in case based reasoning, *Springer-Verlag, London, 2000*.
- Chakrabarti, S. Roy, S. & Soundalgekar, M. V. (2003). Fast and Accurate Text Classification via Multiple Linear Discriminant Projection. *The VLDB Journal The International Journal on Very Large Data Bases*, pp. 170-185.
- Cunningham, P. Nowlan, N. Delany, S. J. & Haahr, M. (2003). A Case-Based Approach in Spam Filtering that Can track Concept Drift. *In The ICCBR'03 Workshop on Long-Lived CBR Systems, Trondheim, Norway*.
- Delany, S.J., Cunningham, P. Tsymbal, A. & Coyle, L. (2004). A Case-Based Technique for Tracking Concept Drift in Spam Filtering. *Journal of Knowledge Based Systems, 18 (4-5), Elsevier*, pp. 187-195.
- Delany, S. J. Cunningham, P. & Coyle, L. (2005). An Assessment of Case-Based Reasoning for Spam Filtering. *Artificial Intelligence Review Journal, Volume 24, Numbers 3-4*, pp. 359-378.
- Flach, P. A., Gyftodimos, E. & Lachiche, N. (2002). Probabilistic Reasoning with Terms. University of Bristol, Loius Pasteur University, Strasbourg.
- Golding A.R., Rosenbloom, P. S. (1996). Improving Accuracy by Combining Rule-Based and Case-Based Reasoning. *Artificial Intelligence 87*, pp. 215-254.
- Han, E. H., Karypis, G., & Kumar, V. (1999). Text Categorisation Using Weight Adjusted k-Nearest Neighbour Classification. Department of Computer Science and Engineering, *Army HPC Research Center, University of Minnesota*.
- Hartley, M., Isa, D., Kallimani, V. P., & Lee, L. H. (2006). A Domain Knowledge Preserving in Process Engineering using Self-Organising Concept. Intelligent System Group, Faculty of Engineering and Computer Science, University of Nottingham Malaysia Campus, Malaysia.
- Isa, D., Prasad, R., Lee, L. H., & Kallimani, V. P. (2007). Oil and Gas Pipeline Acoustic Sensor Monitoring System. Intelligent System Group, Faculty of Engineering and Computer Science, University of Nottingham Malaysia Campus, Malaysia.
- Jacynski, M., Trousse, B. (1998). An Object-Oriented Framework for the Design and the Implementation of Case-Based Reasoners. *In 6th German Workshop on Case-Based Reasoning, Berlin, Germany*.
- Joachims, T. (1998). Text Categorisation with Support Vector Machines: Learning with Many Relevant Features. *In Machine Learning: ECML-98, Tenth European Conference on Machine Learning*, pp. 137--142.

- Kamens, B. (2005). Bayesian Filtering: Beyond Binary Classification. Fog Creek Software, Inc.
- Kontkanen, P., Myllymaki, P., Silander, T., Tirri, H. (1997). A Bayesian Approach for Retrieving Relevant Cases. In *Proceedings of the EXPERSYS-97 Conference*, Sunderland, UK, pp 67--72.
- Kontkanen, P., Myllymaki, P., Silander, T., Tirri, H. (1998). On Bayesian Case Matching. In B. Smyth & P. Cunningham (Eds.), *Advances in case-based reasoning, proceedings of the 4th european workshop (EWCBR-98)*, Vol. 1488, pp. 13--24.
- Lees, B., Corchado, J. (1997). Case-Based Reasoning in a Hybrid Agent-Oriented System, Department of Computing & Information Systems, University of Paisley, Paisley, Scotland.
- McCallum, A., Nigam, K. (2003). A Comparison of Event Models for Naïve Bayes Text Classification. *Journal of Machine Learning Research* 3, pp. 1265-1287.
- Myllymaki, P., Tirri, H. (1993). Bayesian Case-Based Reasoning With Neural Networks. In *Proceeding of the IEEE International Conf. on Neural Networks, San Francisco*.
- Myllymaki, P., Tirri, H. (1994). Massively Parallel Case-Based Reasoning with Probabilistic Similarity Metrics. *Proceedings of the First European Workshop on Case-Based Reasoning*, pp 48--53.
- Nigam, K., Lafferty, J., & McCallum, A. (1999). Using Maximum Entropy for Text Classification. *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pp. 61—67.
- Quinlan, J. R. (1993). *C4.5: Program for Machine Learning*. Morgan Kaufmann, San Mateo, Ca.
- Sahami, M., Dumais, S., Heckerman, D., & Horvits, E. (1998). A Bayesian Approach to Filtering Junk E-Mail. In *AAAI-98 Workshop on Learning for Text Categorisation*.
- Schiaffino, S. N., Amandi, A. (2000). User profiling with Case-Based Reasoning and Bayesian Networks, *IBERAMIA-SBIA 2000 Open Discussion Track*.
- Su, X. (2002). A Text Categorisation Perspective for Ontology Mapping. Department of Computer and Information Science, Norwegian University of Science and Technology, Norway.
- Von Wangenheim, C. G. (2000). Case-Based Reasoning – A Short Introduction, Universidade do Vale do Itajai.
- Weber, R., Aha, D. W., Snadhu, N., Munos-Avila, H. (2001). A Textual Case-Based Reasoning Framework for Knowledge Management Applications. In *Proceedings of the Ninth German Workshop on Case-Based Reasoning*.

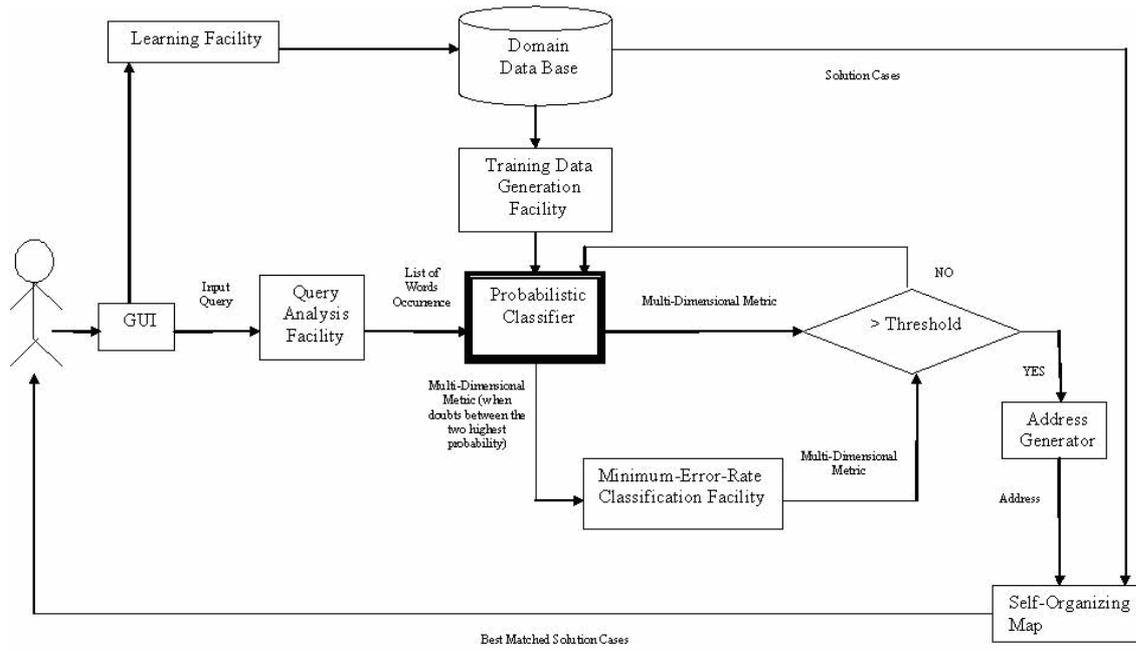


Figure 1. Proposed Enhanced CBR System's Block Diagram

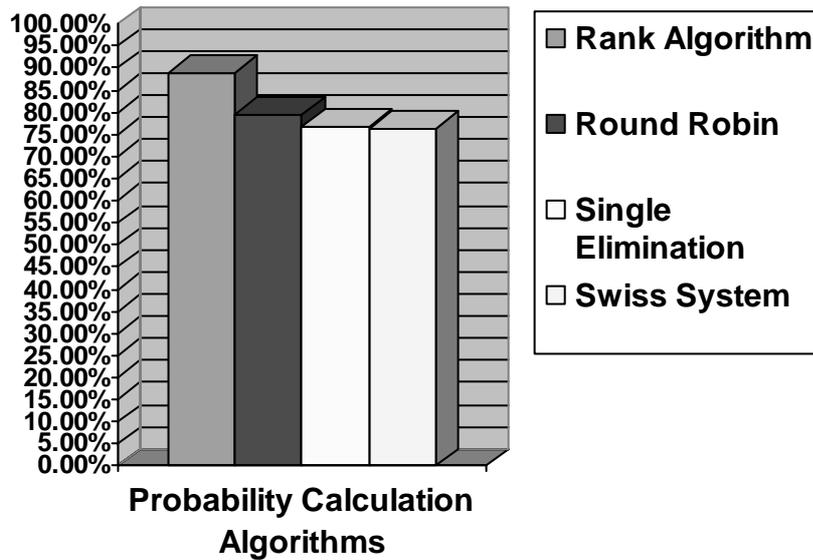


Figure 2. Comparison Chart for Performances of Different Classification Algorithms in Experiment 1

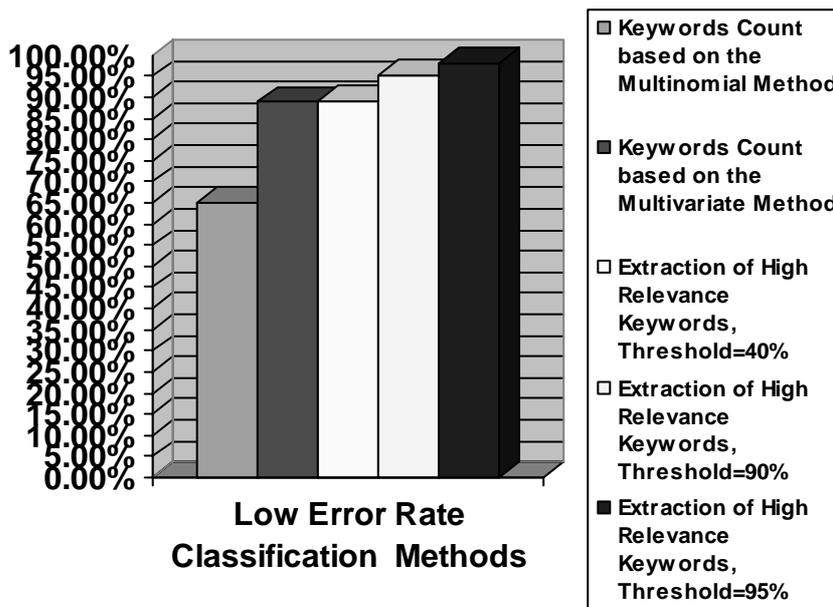


Figure 3. Comparison Chart for Performances of Different Low Error Rate Classification Methods in Experiment 1

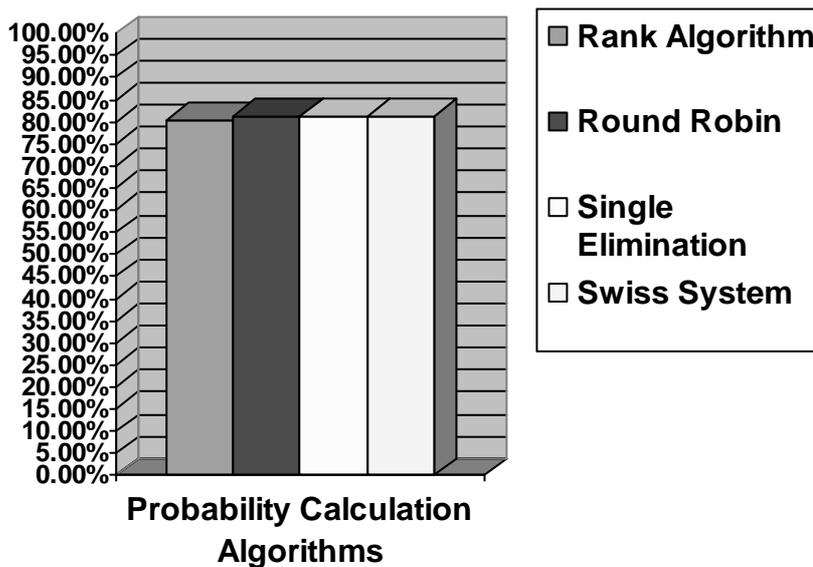


Figure 4. Comparison Chart for Performances of Different Classification Algorithm in Experiment 2

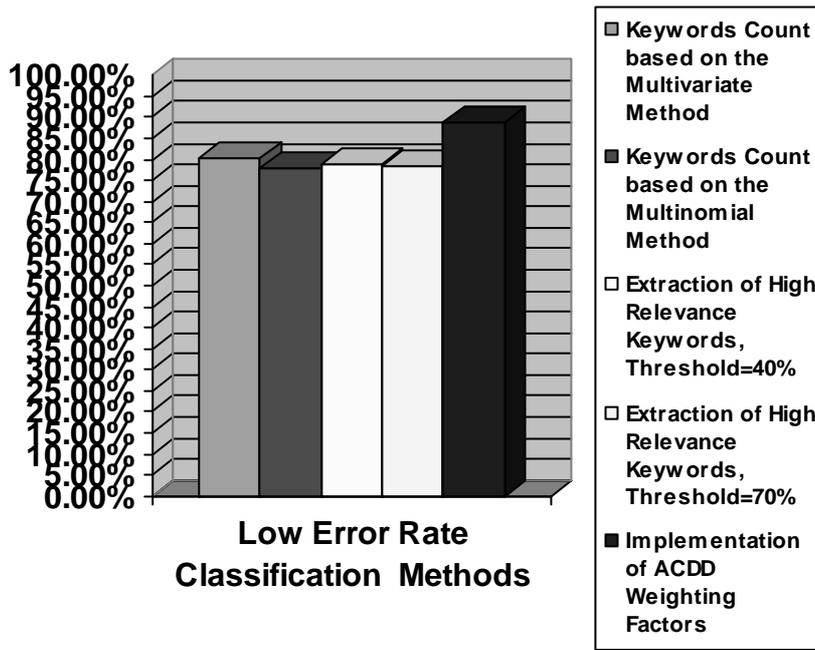


Figure 5. Comparison Chart for Performances of Different Low Error Rate Classification Methods in Experiment 2