A Comparison Study: Web Pages Categorization with Bayesian Classifiers

Zengmei Fu[1], Chuanliang Chen[1], Yunchao Gong[2], Rongfang Bie[1]
[1]*Department of Computer Science, Beijing Normal University, Beijing 100875, China*
[2]*Software Institute, Nanjing University, Nanjing, China*
fzm806@163.com, C.L.Chen86@gmail.com, Corresponding Author: rfbie@bnu.edu.cn

Abstract

In the recent few years, web mining has become a hotspot of data mining with the development of Internet. Web pages classification is one of the essential techniques for web mining since classifying web pages of an interesting class is often the first step of mining the web. The high dimensional text vocabulary space is one of the main challenges of web pages. In this paper, we study the capabilities of bayesian classifiers for web pages categorization. Several feature selection techniques, such as Chi Squared, Information Gain and Gain Ratio are used for selecting relevant words in web pages. Results on benchmark dataset show that the performances of Aggregating One-Dependence Estimators (AODE) and Hidden Naive Bayes (HNB) are both more competitive than other traditional methods.

1. Introduction

Since the Internet has become a huge repository of information, many studies address the issue of web pages classification. It is a fact that web pages are based on loosely structures text and therefore, various statistical text learning algorithms have been applied to web pages categorization [6, 11]. The methods of classification include some novel ones: Naive Bayes, Bayes Network, Hidden Naive Bayes, Aggregating One-Dependence Estimators, Complement class Naive Bayes and some traditional ones such as Support Vector Machine and so on. The origins of our motivation are the great success of Naive Bayes for web pages classification. In this paper, we investigate the capabilities of bayesian algorithms for web pages categorization.

Feature selection means that we want to find a subset of words which help to discriminate between different kinds of web pages. In this paper, we perform several feature selection methods such as *Chi Squared*, *Information Gain* and *Gain Ratio* to extract relevant words of web pages in order to reduce the complexity of classifiers and preserve their performances.

The remainder of this paper is organized as follows. In section 2, we briefly review the five bayesian classification methods. Section 3 describes several feature selection methods. In section 4, we demonstrate performance measures, experiments' results and analyze. Finally, we conclude our work in Section 5.

2. Comparison of Different Classifiers

2.1. Naive Bayes

The Naive Bayesian classifier is also simply named Naive Bayes [1, 2, 3, 4, 6]. It is widely deployed for classification due to its simplicity, efficiency and efficacy.

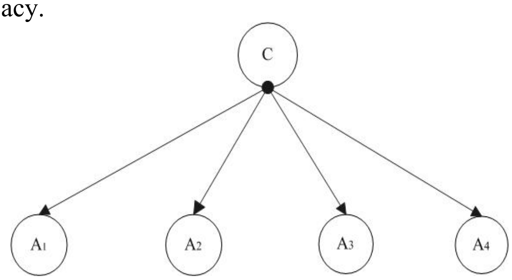

Figure 1. An example of Naive Bayes

The structure of Naive Bayes is depicted in Fig. 1. In Naive Bayes, each attribute node has the class node as its parent but it does not have any parent from attribute nodes.

For a given module sample, Naive Bayes classifier searches for a class c_i which maximizes the posterior probability $P(ci|x;\theta')$ through applying the bayes rule. Then x can be classified by computing the equation as follows:

$$c_l = \arg\max_{c_i \in C} P(c_i \mid \theta')P(x \mid c_i;\theta') \quad .$$

2.2. Bayes Network

By specifying a set of conditional independence statements together with a set of conditional probability functions, Bayes Network [3] estimates the probability density function governing a set of random variables.

Assume that A_1, A_2,…, A_n are n attributes. E is an example which represented by a vector (a_1, a_2, …, a_n), where a_i is the value of A_i. The class variable is represented by C. We use c to represent the value that C takes and $c(E)$ to denote the class of E. The definition of Bayes Network is represented by the follow equation.

$$c(E) = \arg\max_{c \in C} P(c)P(a_1, a_2, ...a_n \mid c) \quad .$$

2.3. Hidden Naive Bayes

The following picture shows the structure of Hidden Naive Bayes (HNB).

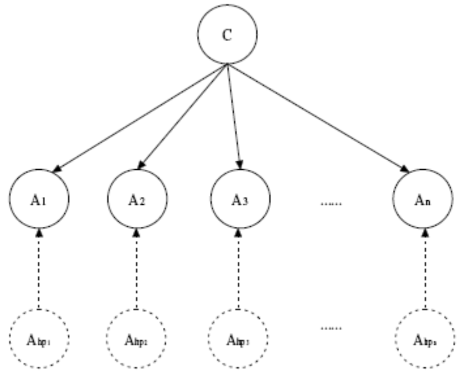

Figure 2. The structure of HNB

In an HNB [3], hidden parents of attributes represent attribute dependencies. The class node is represented by C which is also the parent of all attribute nodes. Each attribute A_i has a hidden parent A_{hpi}, where i = 1, 2,…, n, represented by a dashed circle. In order to distinguish from regular arcs, the arc from the hidden parent A_{hpi} to A_i is also represented by a dashed directed line.

The follow equation defines the joint distribution represented by an HNB.

$$P(A_1, ...A_n, C) = P(c)\prod_{i=1}^{n} P(A_i \mid A_{hpi}, C) \quad ,$$

where $P(A_i \mid A_{hpi}, C) = \sum_{j=1, j\neq i}^{n} W_{i,j} * P(A_i \mid A_j, C) \quad .$

Essentially, the hidden parent A_{hpi} for A_i is a mixture of the weighted influences from all other attributes.

HNB can be defined as follow equation and M is an example.

$$c(M) = \arg\max_{c \in C} P(c)P(a_i \mid a_{hpi}, c) \quad .$$

```
Algorithm HNB (X)
Input: a set X of training web pages.
For each value c of C
    Compute P(c) from X
For each pair of words/attributes $A_i$ and $A_j$
    For each assignment $a_i$, $a_j$, and c to $A_i$, $A_j$, and C
        Compute P ($a_i$; $a_j$|c) from X
For each pair of attributes $A_i$ and $A_j$
    Compute $I_P (A_i; A_j | C)$
For each attribute $A_i$
    Compute $W_i = \sum_{j=1, j\neq i}^{n} I_p(A_i; A_j \mid C)$ ,
    For each attribute $A_j$ and $j \neq i$
        Compute $W_{ij} = \frac{I_p(A_i; A_j \mid C)}{W_i}$ ,
Output: HNB models for X
```

Figure 3. The process of HNB algorithm

2.4. AODE

Abbreviation of Aggregating One-Dependence Estimators (AODE）intends to average models from a restricted class of one-dependence classifiers, the class of such classifiers have all other attributes depend on a common attribute and class [4].

Selecting a limited class of one-dependence classifiers in the process of AODE make NB's attribute independence assumption weaken. The selected class is one-dependence classifiers and the parent of all other attributes is a single attribute [4]. When it comes to classifying an object x = <x_1,…, x_n>, the models in which the training data contain fewer than m samples of the value for x of the parent attribute x_i can be excluded by AODE and m is a predefined threshold parameter on the size of samples for statistical inference purposes. The main equation is defined as follows:

$$p(y, \mathrm{x}) = \frac{\sum_{i:1\leq i\leq n \wedge F(x_i)\geq m} P(y, x_i)P(\mathrm{x} \mid y, x_i)}{|\{i : 1 \leq i \leq n \wedge F(x_i) \geq m\}|} \quad , \qquad (1)$$

where $F(x_i)$ is a count of the number of training instances having attribute-value x_i and is used for enforcing the limit m.

AODE estimates the class probabilities through the above equation. Since the denominator of Eq. 1 is invariance and it need not to be calculated, we can find a new way to estimate the probabilities of Eq. 1 and to seek the class which maximizes the obtained term. The class satisfies:

$$\arg\max_{y}\left(\sum_{i:1\le i\le n\wedge F(x_i)\ge m}\hat{P}(y,x_i)\prod_{j=1}^{n}\hat{P}(x_j\mid y,x_i)\right) ,$$

When $\neg\exists i:1\le i\le n\wedge F(x_i)\ge m$, AODE defaults to NB [4].

2.5. Complement class Naive Bayes

In order to deal with skewed training data, we introduce a method called Complement class Naive Bayes (CNB) [7] which is a complement class of Naive Bayes. CNB estimates parameters using data from all classes except *c*. Due to using a more even amount of training data which could lessen the bias in the weight estimates, the result of CNB get more stable weight estimates and improved classification accuracy. The estimate of CNB is as follows:

$$\hat{\theta}_{ci}=\frac{N_{ci}+\alpha_i}{N_c+\alpha} ,$$

where $N_{\overline{ci}}$ is the number of times that word *i* occurred in documents of classes other than *c*; $N_{\overline{c}}$ is the total number of word occurrences in classes other than *c*; α_i and α are smoothing parameters. The rule of classification is defined as follows:

$$l_{CNB}(d)=\arg\max_c[\log p(\vec{\theta}_c)-\sum_i f_i\log\frac{N_{ci}+\alpha_i}{N_c+\alpha}] .$$

3. Feature Selection Methods

There are four feature selection methods [13-15] used in this paper to evaluate the performances of algorithm. The definitions are stated below.

In the following equations, *m* denotes the number of classes, and C_i represents the *i*th class. The number of partitions which a feature could split the training set into is represented by *V*. N_{c_i} is the total number of samples in class *i*, where *N* is the total number of samples. The number of samples belongs to class *i* in the *v*th partition denoted by $N_{c_i}{}^{(v)}$.

Chi Squared: The statistic of *Chi Squared* is calculated by comparing the obtained frequency with the priori frequency of the same class. The definition is as follows:

$$\chi^2=\sum_{i=1}^{m}\sum_{v=1}^{V}\frac{(N_{C_i}{}^{(v)}-\tilde{N}_{C_i}{}^{(v)})^2}{\tilde{N}_{C_i}{}^{(v)}} ,$$

where, the prior frequency of the class is denoted by the equation: $\tilde{N}_{C_i}{}^{(v)}=(N^{(v)}/N)N_{C_i}$.

Information Gain: *Information Gain* is based on the feature's impact about decreasing entropy and it can be defined with the follow equation:

$$InfoGain=[\sum_{i=1}^{m}-(\frac{N_{C_i}}{N})\log(\frac{N_{C_i}}{N})]-[\sum_{v=1}^{V}(\frac{N^{(v)}}{N})\sum_{i=1}^{m}-(\frac{N_{C_i}}{N})\log(\frac{N_{C_i}}{N^{(v)}})] \cdot$$

Gain Ratio: *Gain Ratio* is first used in C4.5 and the definition is as follows:

$$GainRatio=InfoGain/[\sum_{i=1}^{m}-(\frac{N_{C_i}}{N})\log(\frac{N_{C_i}}{N})] .$$

Symmetrical Uncertainty: *Symmetrical Uncertainty* has been described in books about information theory and in numerical recipes [19]. It is often used as a fast correlation measure to evaluate the relevance of individual features. With this method, the most relevant feature is positioned at the beginning of the list. This criterion compensates for the inherent bias of *Information Gain* through dividing it by the sum of the entropies of class labels *M* and features *X* [15]:

$$SU=2\times InfoGain/(Ent(M)+Ent(X)) ,$$

The value of SU ranges from 0 to 1. A value 0 indicates the attribute *X* and the class *M* have no association while a value 1 indicates *X* can completely predict *M*.

4. Experiments

4.1. Corpus and Preprocessing

In our experiments, we use CMU industry sector which is a collection of web pages belonging to companies from various economic sectors. A subset of the original data which form a two-level hierarchy is used in this research. There are 527 instances partitioned into seven classes: materials, energy, financial, healthcare, technology, transportation and utilities.

Each web page of the corpus is represented as a set of words. After analyzing all the web pages of a corpus, a dictionary with *N* words is formed. Two data types are used in the experiments, one is Boolean and the other is term frequencies (TF). The type of Boolean represents whether a word occurs in web pages while TF describes the frequency of a word in web pages. During preprocessing, we perform word stemming, stop-word removable and Document Frequency Thresholding (DFT) [18], all of them are used for reducing the dimension of feature space for web pages categorization. In the end, the first 3,000 tokens of dictionary are extracted according to their Mutual Information and form the corpus used in this paper.

4.2. Performance Measure

Accuracy and *F*-measure are two popular evaluation metrics [17] of text categorization domain used for measuring the performance of classifiers.

Accuracy: Accuracy represents the percentage of correct predictions in total predictions. It usually can be defined as follows:

$$Accuracy = \frac{P_c}{P_t} \times 100\% \quad ,$$

where P_c depicts the number of correct predictions and P_t is the number of total predictions.

***F*-measure**: *F*-measure can be defined as follows:

$$F = \frac{2R \times P}{R + P} \quad ,$$

where Recall is represented by *R* and it is the percentage of the messages for a given category which are classified correctly; *P* is the Precision, the percentage of the predicted messages for a given class which are classified correctly. *F*-measure ranges from 0 to 1 and the higher the better.

4.3. Results and Analysis

We choose 10-fold cross-validation on this benchmark dataset to estimate the performances of classification in our experiments, studying the comparison of the above eight different methods and four feature selection methods.

When it comes to Fig. 4, we select top 100 relevant words by performing the four feature selection methods and compare the capabilities of the above eight algorithms. The results of our experiments show that HNB is a better classifier than the other seven methods both evaluated by accuracy and *F*-measure in these two figures. In Fig. 4, the accuracy of HNB reach to 88.97% and *F*-measure hit 0.895, both of them are the highest. We also find that selecting relevant words by SU is more competitive than other three ones since both the highest accuracy and *F*-measure occurs when we select features according to SU scores.

Since the poor research on SU for web pages categorization, we further study the capability and stability of SU by performing classifiers on different number of relevant words selected according to SU scores.

In the following experiment, we sort words according to their SU scores, and then study the performances of the above eight classifiers on different number of top relevant words. As is showed in Fig. 5, we select number of attributes through removing top *N* words according to SU scores, where *N* is the number of attributes and in our experiments, it

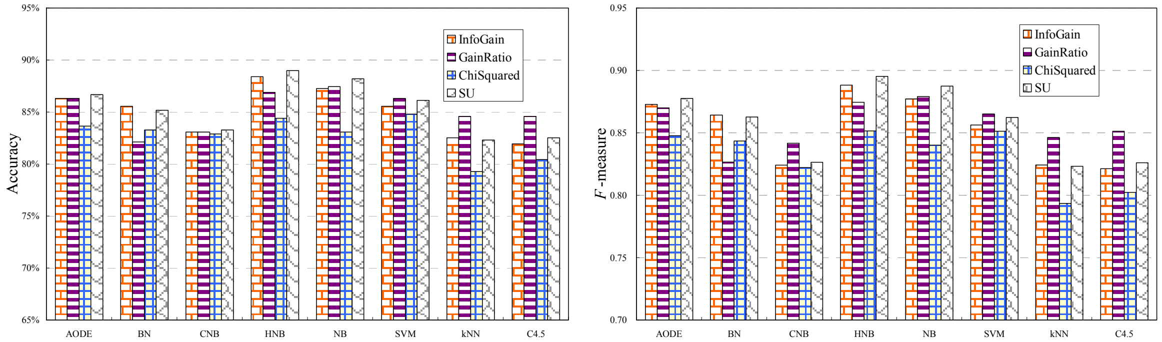

Figure 4. The accuracy (left) and *F*-measure (right) comparison of Bayesian classifiers with the Boolean data type, BN represents Bayes Network, NB represents Naive Bayes.

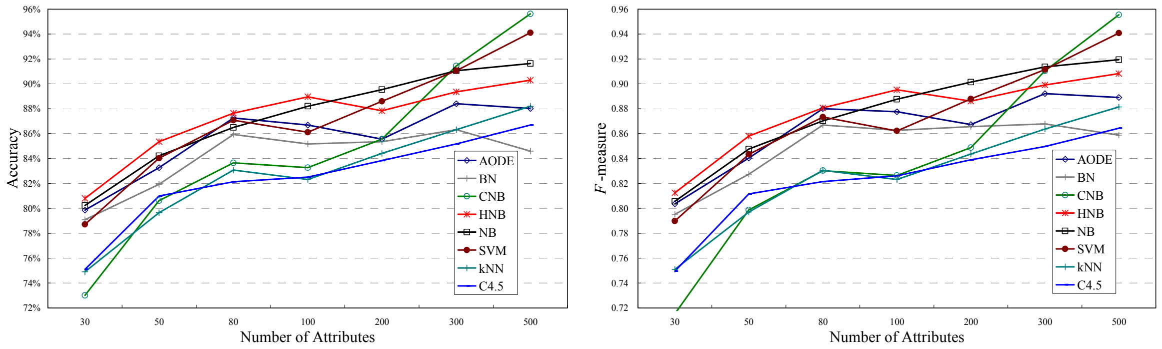

Figure 5. Accuracy (left) and *F*-measure (right) curves of the eight classifiers with different numbers of relevant words according to their SU scores and Boolean data type, BN represents Bayes Network, NB represents Naive Bayes.

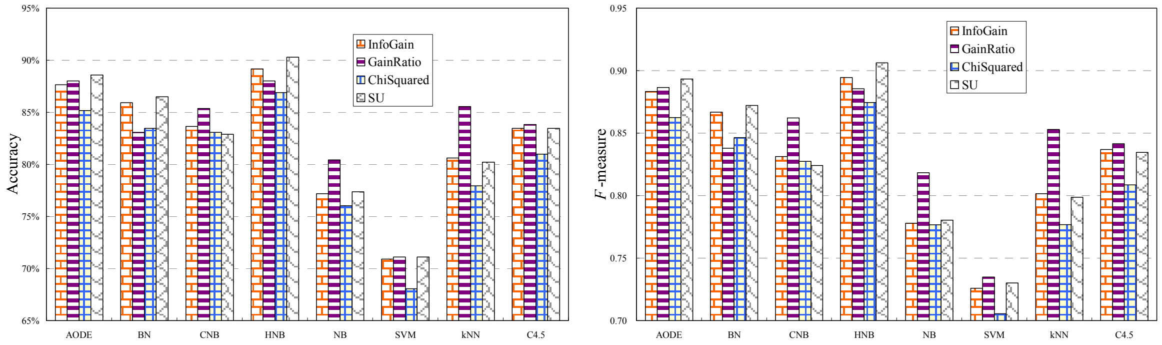

Figure 6. The accuracy (left) and *F*-measure (right) comparison of Bayesian classifiers with the TF data type, BN represents Bayes Network, NB represents Naive Bayes.

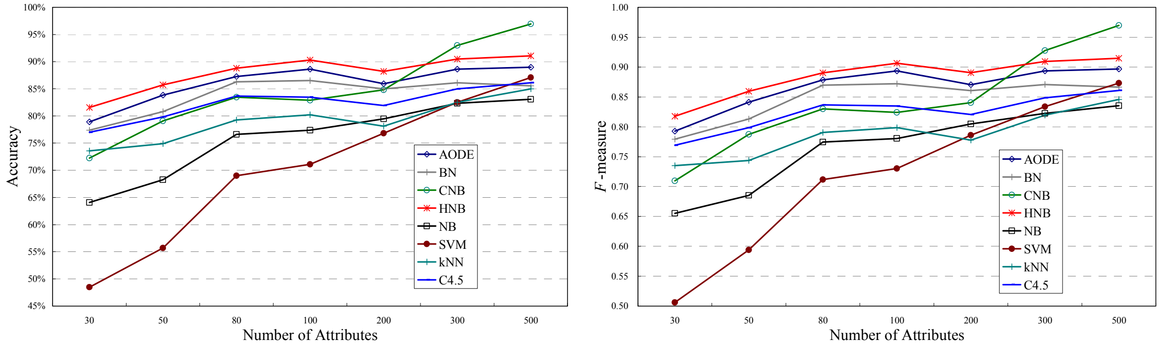

Figure 7. Accuracy (left) and *F*-measure (right) curves of the eight classifiers with different numbers of relevant words according to their SU scores and TF data type, BN represents Bayes Network, NB represents Naive Bayes.

range from 30 to 500. It is easy to find that if the number of attributes reaches to 500, CNB will achieve the highest accuracy and the highest *F*-measure, 95.63% and 0.956 respectively. Compared with KNN and C4.5, bayesian classifiers perform satisfying, especially for HNB and AODE which are more stable than other classifiers both in accuracy and *F*-measure. Since the high complexity, SVM is time-consuming and it is more comfortable to small-size problems. In the contrast, the complexity of AODE or HNB is much little, so they are more feasible in practice. Certainly, the ability of SU for relevant words selection is also promising.

Similar as Boolean, we study of TF as follows. In Fig. 6, we also select top 100 relevant words. The results obviously show that both accuracy and *F*-measure of SVM descend while AODE and HNB always perform well. From Fig. 6, we observe the accuracy of HNB reach 90.30%, which is the highest. Similarly, the highest *F*-measure is also achieved by HNB which is 0.906.

The lowest accuracy and *F*-measure is 68.06% and 0.706 respectively, which are both achieved by SVM. Also, we find that selecting relevant words by SU is more competitive since both the highest accuracy and *F*-measure occurs when we select feature according to SU scores.

In the following experiment, we sort the words according to their SU scores, and then study the performances of the eight classifiers on different number of top *N* relevant words. In this experiment, *N* ranges from 30 to 500. In Fig. 7, we find if the number of attributes reaches to 500 then the accuracy of CNB will hit the highest value 96.96%. Fig. 7 also shows the changes of *F*-measure by performing different classifiers and we find the *F*-measure of CNB will reach to the highest value when the number of attributes is 500.

Our experiments also show the comparison across the above two data types. On the Boolean data type, we find 87.34% is the highest average accuracy which is achieved by NB and the second one is 87.18%, which is achieved by HNB. SVM is the third highest and AODE is the fourth, the values of them are 87.10% and 85.58% respectively. As to the average *F*-measure, 0.878 is the highest which is achieved by NB. Second to it, the average *F*-measure of HNB is 0.877. The third one is 0.873, which is achieved by SVM. 0.864 is the fourth highest and it is achieved by AODE. From the data offered above, we believe

Bayesian classifiers perform better than others in web pages classification.

Moreover, comparing with Boolean, the highest average accuracy on TF is 88.02%, which is achieved by HNB and AODE is the second one with the data of 86.01%. In contrast, NB and SVM perform not so well as on Boolean. The average accuracy of NB is 75.88% and SVM is 70.10%, which are the third and the fourth highest. As to the average *F*-measure, the highest one is 0.884, which is achieved by HNB. The second one is AODE with the value of 0.867. NB is the third highest and SVM is the fourth, the values of them are 0.765 and 0.719 respectively. Across the two types, we find both the highest average accuracy and *F*-measure are achieved by HNB on TF. All the evidences offered above show that TF contains more information than Boolean, and classifiers perform better both in the average accuracy and the average *F*-measure with TF data type.

5. Conclusion

In this paper, we report our work on web pages categorization and the comparison of bayesian classification methods: Naive Bayes, Bayes Network, AODE, HNB and CNB. Other traditional methods are also performed for comparison. In our experiments, several feature selection methods such as *Chi Squared*, *Information Gain* and *Gain Ratio* are used for selecting relevant words in web pages. Two popular evaluation metrics, accuracy and *F*-measure are used for evaluating the performances of classifiers. Our empirical study shows the abilities of bayesian classifiers perform satisfying, especially for AODE and HNB which are both more competitive than other methods. Also, SVM performs well in certain number of attributes although limited to its high complexity.

6. Acknowledgement

The research work described in this paper was supported by the National Science Foundation of China under the Grant No. 10601064.

References

[1] Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. “A bayesian approach to filtering junk e-mail”. *In Learning for Text Categorization*, Madison, Wisconsin, 1998.

[2] V. Metsis, I. Androutsopoulos, G. Paliouras. “Spam Filtering with Naive Bayes – Which Naive Bayes?” *CEAS 2006 Third Conference on Email and Anti-Spam*, Mountain View, California USA*,* Jul. 2006.

[3] H. Zhang, L. Jiang, J. Su. “Hidden Naive Bayes” *Proceedings of the Twentieth National Conference on Artificial Intelligence*. pp. 919-924, AAAI Press, 2005.

[4] Geoffrey I. Webb, Janice R. Boughton, Zhihai Wang. “Not So Naive Bayes: Aggregating One-Dependence Estimators”. *Machine Learning,* 58, 5–24, 2005.

[5] J. Ross Quinlan. “Induction of Decision Trees”. *Machine Learning*, 1:81-106, 1986.

[6] H. Mase. “Experiments on Automatic Web Page Categorization for IR System”. *Technical Report*, Stanford University, Stanford, Calif. 1998.

[7] Jason D. M. Rennie, Lawrence Shih, Jaime Teevan, David R. Karger. “Tackling the Poor Assumptions of Naive Bayes Text Classifiers”. *Proceedings of the Twentieth International Conference on Machine Learning* (ICML-2003), Washington DC, 2003.

[8] Ross Quinlan. “C4.5: Programs for Machine Learning,” *Morgan Kaufmann Publishers*, San Mateo, CA, 1993.

[9] Industry Sector Dataset http://www.cs.cmu.edu/~TextLearning/datasets.html 2005.

[10] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy. “Improvements to Platt's SMO Algorithm for SVM Classifier Design”. *Neural Computation*, 13(3), pp 637-649, 2001.

[11] Zheng Zhao, Huan Liu. “Searching for Interacting Features”. *In: Proc. of International Joint Conference on Artificial Intelligence*, Hyderabad, India, Jan, 2007.

[12] Hwanjo Yu, Jiawei Han, Kevin Chen-Chuan Chang. “PEBL: Web Page Classification without Negative Examples”. *IEEE Trans. Knowl.* Data Eng. 16(1): 70-81, 2004.

[13] Kira, K, L. A. Rendell. “The Feature Selection Problem: Traditional Methods and New Algorithm”. *Proceedings of AAAI'*92, 1992.

[14] Kira, K, L. A. Rendell. “A Practical Approach to Feature Selection”. D.Sleeman and P.Edwards (eds.): Machine Learning: *Proceedings of International Conference*. pp. 249–256, Morgan Kaufmann, 1992.

[15] M.A. Hall, L.A. Smith. “Practical feature subset selection for machine learning”, *In Proceedings of the 21st Australian Computer Science Conference,* pp. 181–191, 1998.

[16] R. C. Holte. “Very Simple Classification Rules Per-form Well on Most Commonly Used Datasets,” *Machine Learning*, vol. 11, pp. 63-91, 1993.

[17] G. Buddhinath, D. Derry. “A Simple Enhancement to One Rule Classification,” *Technique Report at http://goanna.cs.rmit.edu.au/~gjayatil/OtherLinks/Extra*.php, 2006.

[18] Stemming:http://www.comp.lancs.ac.uk/computing/research/stemming/general/, Access 2006.

[19] L. Yu, H. Liu. “Feature selection for high-dimensional data: a fast correlation-based filter solution”. *In Proceedings of ICML 2003*, 2003.