Feature selection for unbalanced class distribution and Naive Bayes

Dunja Mladenić
Department of Intelligent Systems
J.Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Dunja.Mladenic@ijs.si

Marko Grobelnik
Department of Intelligent Systems
J.Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Marko.Grobelnik@ijs.si

Abstract

This paper describes an approach to feature subset selection that takes into account problem specifics and learning algorithm characteristics. It is developed for the Naive Bayesian classifier applied on text data, since it combines well with the addressed learning problems. We focus on domains with many features that also have a highly unbalanced class distribution and asymmetric misclassification costs given only implicitly in the problem. By asymmetric misclassification costs we mean that one of the class values is the target class value for which we want to get predictions and we prefer false positive over false negative. Our example problem is automatic document categorization using machine learning, where we want to identify documents relevant for the selected category. Usually, only about 1%-10% of examples belong to the selected category. Our experimental comparison of eleven feature scoring measures show that considering domain and algorithm characteristics significantly improves the results of classification.

1 Introduction

In propositional learning a problem domain is given by a set of examples, where each example is described with a class value and a vector of feature values. Features used to describe examples are not necessarily all relevant and beneficial for inductive learning. Additionally, a high number of features may slow down the induction process while giving similar results as obtained with a much smaller feature subset. Feature subset selection is commonly used when learning on text data, since most text domains are characterized by several tens of thousands of features. There is a growing interest in the usage of machine learning on text data. It is especially stimulated by the intensive growth of the World Wide Web that can be seen as a widely accessible, large, distributed source of text data. The problem of automatic document categorization is well known in information retrieval and usually tested on publicly available data bases (eg. Reuters, MEDLINE). Here we use machine learning for document categorization using one of the existing Web hierarchies named Yahoo (Filo and Yang, 1997). In this way, current situation on the Web captured in Web hierarchy can be used for automatic document categorization.

Section 2 gives description of feature subset selection as used in machine learning and in text-learning. Section 2.1 describes several known and some new feature scoring measures applied on text data. Data characteristics and classification algorithm are described in Section 3. Experimental comparison of the tested measures on real-world data is given in Section 4. The results are discussed in Section 5.

2 Feature subset selection approaches

According to John et al. (John et al., 1994) there are two main approaches to feature subset selection used in machine learning: filtering and wrapper. In the filtering approach a feature subset is selected independently of the learning method that will use the selected features. The idea of the wrapper approach is to select feature subset using the evaluation function based on the same algorithm that will be used for learning. This can result in a rather time consuming process. In information retrieval and also in text-learning the whole process of feature subset selection is simplified with the assumption of feature independence. In this way the solution quality is traded for the time needed to find a solution, justified by the large number of features usually present in text domains.

2.1 Feature selection on text data

The usual way of learning on text defines a feature for each word that occurred in the training documents. This can easily result in several tens of thousands of features. Here we use the extended representation that defines a feature for each word sequence containing up to 5 consecutive words (Mladenić and Grobelnik, 1998). Methods for feature subset selection that are used on text are very simple compared to the methods developed in machine learning. Basically, some evaluation function that is applied to a single feature is used. All the features are independently evaluated, a score is assigned to each of them and the features are sorted according to the assigned score. Then, a predefined number of the best features is taken to form the solution feature subset.

Scoring of individual features can be performed using some of the measures used in machine learning for instance, *Information gain* used in decision tree induction (Quinlan, 1993). In our comparison eleven feature scoring measures were included. Information gain was included as the well known measure successfully used in some text-learning experiments. *Expected cross entropy* used in text-classification experiments (Koller and Sahami, 1997) is similar to Information gain. The difference is that instead of calculating average over all possible feature values, only the value denoting that word occurred in a document is considered. Our experiments show that this means an important difference in the resulting performance. *Mutual information* used in text-classification experiments (Yang and Pedersen, 1997) is similar to *Cross entropy for text* with the latter additionally taking into account word probability. These two measures and a very simple frequency measure proposed in (Yang and Pedersen, 1997) were reported to work well on text data. This third measure is calculated either as the number of documents that contain word W referred to as document frequency $DF(W)$ or as the number of occurrences of word W referred to as *Term frequency* $TF(W)$. This measure requires the stop words removal. We use the second definition: $Freq(W) = TF(W)$. Inspired by the way Cross entropy for text is derived from Information gain, we propose *Weight of evidence for text* that is based on the average absolute weight of evidence used in machine learning (Kononenko, 1995). *Odds ratio* is commonly used in information retrieval, where the problem is to rank out documents according to their relevance for the positive class value using occurrence of different words as features (van Rijsbergen, Harper and Porter, 1981). Our experiments show that this measure is especially suitable to be used in a combination with the Naive Bayesian classifier for our kind of problems. We propose four variants of *Odds ratio*, to test if the results are sensitive to some modifications in the formula. As a baseline method we used random scoring method defined to score each word by a random number. Table 1 contains formulas of the used scoring measures.

3 Domain and algorithm characteristics

We perform experiments on five domains defined from the Yahoo hierarchy as described in (Mladenić, 1998). Each domain is given as a classification hierarchy of text documents, with the more general categories closer to the root of the hierarchy. Each category is denoted by keywords, describing category content. More specific category is named by adding a keyword to the name of the more general category that is one level higher in the hierarchy. Each document is represented here as a feature vector, where a feature represents word or word sequence. Features are generated using an algorithm (Mladenić and Grobelnik, 1998) similar to the Apriori algorithm used for association rule generation. Five domains with the following characteristics are used in our experiments. *'Entertainment'* having 8,081 class values (nodes in the hierarchy), 30,998 features (selected word sequences) and 79,011 examples (actual Web documents). *'Arts and Humanities'* having 3,085 class values, 11,473 features and 27,765 examples. *'Computers and Internet'* having 2,652 class values, 7,631 features and 23,105 examples. *'Education'* having 349 class values, 3,215 features and 5,406 examples. *'References'* having 129 class values, 923 features and 1,995 examples. In order to learn from the class hierarchy and handle a large number of class values we induce a binary classifier for each class value (each of the hierarchy nodes, leaf or non-leaf) by collecting word probabilities from all the documents in and below the node (see (Mladenić, 1998), (Mladenić, 1998a) for more information). These documents are positive examples while all the other documents in the hierarchy are used as negative. This means that the most classifiers are induced from highly unbalanced class distribution with only about 1%-10% of posi-

$$
\begin{array}{ll}
InfGain(W) = & P(W)\sum_i P(C_i|W)\log\frac{P(C_i|W)}{P(C_i)} + P(\overline{W})\sum_i P(C_i|\overline{W})\log\frac{P(C_i|\overline{W})}{P(C_i)} \\
CrossEntropyTxt(W) = & P(W)\sum_i P(C_i|W)\log\frac{P(C_i|W)}{P(C_i)} \\
MutualInfoTxt(W) = & \sum_i P(C_i)\log\frac{P(W|C_i)}{P(W)} \\
Freq(W) = & TF(W) \\
WeightOfEvidTxt(W) = & \sum_i P(C_i)\times P(W)\times|\log\frac{P(C_i|W)(1-P(C_i))}{P(C_i)(1-P(C_i|W))}| \\
OddsRatio(W) = & \log\frac{P(W|pos)(1-P(W|neg))}{(1-P(W|pos))P(W|neg)} \\
WeightedOddsRatio(W) = & P(W)\times OddsRatio(W) \\
LogProbRatio(W) = & \log\frac{P(W|pos)}{P(W|neg)} \\
ExpProbDiff(W) = & e^{P(W|pos)-P(W|neg)} \\
CondOddsRatio(W) = & 0.1\times\log\frac{P(W|neg)(1-P(W|pos))}{(1-P(W|neg))P(W|pos)} \text{ if } P(W|neg)-P(W|pos)>0.9 \\
 & \text{and, otherwise equals } OddsRatio(W)
\end{array}
$$

Table 1: Formulas of the used feature scoring measures. Where $P(W)$ is the probability that word W occurred, $\overline{W}$ means that word doesn't occur, $P(C_i)$ in the probability of the i-th class value, $P(C_i|W)$ is the conditional probability of the i-th class value given that word W occurred, $P(W|C_i)$ is the conditional probability of word occurrence given the i-th class value, $P(W|pos)$ is the conditional probability of word W occurring given the class value 'positive', $P(W|neg)$ is the conditional probability of word W occurring given the class value 'negative', $TF(W)$ is term frequency (the number of times word W occurred).

tive examples. Since each classifier competes with the other classifiers, it is important that a classifier identifies documents belonging to its category. Thus we say that our problem has asymmetric misclassification costs given only implicitly in the problem. By asymmetric misclassification costs we mean that one of the class values (positive) is the target class value for which we want to get predictions and we prefer false positive over false negative. Problem with similar characteristics addressed in (Mladenić, 1998b) is predicting relevant/interesting documents on the Web, based on the clicked hyperlinks (that can be used by a personal browsing assistant).

Experiments are performed using the Naive Bayesian classifier base on the multinomial event model as suggested in (McCallum and Nigam, 1998). Notice that the product goes over all word sequences that occur in the representation of document *Doc*.

$$P(C|Doc) = \frac{P(C)\Pi_{W_j\in Doc}P(W_j|C)^{TF(W_j,Doc)}}{\sum_i P(C_i)\Pi_{W_l\in Doc}P(W_l|C_i)^{TF(W_l,Doc)}}$$

Where $P(W_j|C)$ is the conditional probability here estimated using Laplace probability estimate, $TF(W_j, Doc)$ is the frequency of word W_j in document *Doc*. Notice that using word sequences instead of only single words means even stronger violation of the feature independence assumption used in the Naive Bayesian classifier.

4 Experimental results

In our experiments we compare different feature scoring measures and observe the influence of the number of selected features to the system performance. Since we have a set of classifiers, the number of selected features is determined relatively to the classifier category size expressed by the number of features in positive examples. We refer to this relative number of features as *Vector size*. In this way, a classifier for a larger category is using more features than a classifier for some smaller category, while both classifying the same testing example. Reported results are averaged over 5 repetitions using hold-out testing on independent set of 500 (300 for the two smaller domains) randomly selected testing examples. In order to enable operational usage of the system on larger domains we include the pruning mechanisms described in (Grobelnik and Mladenić, 1998). Result is speed up of the classification process, since for each document classification, not all but only promising categories are considered (85%-95% of all categories are pruned).

To evaluate the results we use Precision, Recall and F_2-measure as commonly used evaluation measures for text data (Lewis, 1995). $F-measure$ is a combination of Precision P and Recall R commonly used in information retrieval $F_\beta = \frac{(1+\beta^2)P\times R}{\beta^2 P+R}$. The relative importance of each is expressed with the value of parameter β. We report average Precision and Re-

Dom. name	Scoring measure	Average on keyword assignment F1-measure	F2-measure	Precision	Recall
Ent.	**Odds ratio**	**0.48** ± 0.006	**0.59** ± 0.007	**0.44**± 0.006	**0.80** ± 0.006
	Weight of evidence for text	**0.48** ± 0.002	0.52 ± 0.004	0.74 ± 0.003	0.58 ± 0.003
	Term frequency	0.39 ± 0.003	0.49 ± 0.12	0.41± 0.006	0.71 ± 0.010
	Cross entropy Txt	0.29 ± 0.007	0.39 ± 0.007	0.35± 0.003	0.69 ± 0.007
	Mutual information Txt	0.25 ± 0.005	0.27 ± 0.006	0.57± 0.007	0.38 ± 0.007
	Information gain	0.27 ± 0.008	0.22 ± 0.004	0.86± 0.005	0.21 ± 0.006
	Random	0.002± 0.001	0.002± 0.001	0.99± 0.006	0.001± 0.007
Arts.	**Odds ratio**	0.46 ± 0.003	**0.59** ± 0.005	**0.40**± 0.002	**0.83** ± 0.006
	Term frequency	**0.47** ± 0.007	**0.58** ± 0.008	**0.48**± 0.003	**0.77** ± 0.009
	Weight of evidence Txt	0.50 ± 0.012	0.56 ± 0.012	0.70 ± 0.001	0.62 ± 0.012
	Cross entropy Txt	0.32 ± 0.003	0.44 ± 0.004	0.33± 0.004	0.75 ± 0.008
	Mutual information Txt	0.31 ± 0.005	0.35 ± 0.004	0.56± 0.007	0.46 ± 0.006
	Information gain	0.25 ± 0.006	0.21 ± 0.005	0.94± 0.002	0.20 ± 0.004
	Random	0.0051± 0.001	0.001± 0.001	0.99± 0.001	0.001± 0.001
Comp.	**Odds ratio**	0.46 ± 0.006	**0.60** ± 0.006	**0.40**± 0.007	**0.84** ± 0.005
	Term frequency	**0.48** ± 0.008	**0.58** ± 0.007	**0.50**± 0.004	**0.74** ± 0.005
	Weight of evidence Txt	0.46 ± 0.002	0.51 ± 0.11	0.76± 0.007	0.56 ± 0.010
	Cross entropy Txt	0.37 ± 0.007	0.49 ± 0.008	0.35± 0.004	0.75 ± 0.007
	Mutual information Txt	0.36 ± 0.003	0.38 ± 0.003	0.62± 0.005	0.45 ± 0.005
	Information gain	0.21 ± 0.005	0.17 ± 0.005	0.94± 0.005	0.15 ± 0.005
	Random	0.01 ± 0.001	0.01 ± 0.001	0.99 ± 0.001	0.004± 0.001
Edu.	**Term frequency**	**0.48** ± 0.007	**0.55** ± 0.007	**0.57**± 0.010	**0.65** ± 0.010
	Odds ratio	0.33 ± 0.008	**0.48** ± 0.008	**0.36**± 0.010	**0.81** ± 0.005
	Mutual information Txt	0.40 ± 0.004	0.46 ± 0.007	0.48± 0.010	0.59 ± 0.010
	Cross entropy Txt	0.32 ± 0.009	0.46 ± 0.007	0.28± 0.010	0.82 ± 0.006
	Weight of evidence Txt	0.30 ± 0.007	0.34 ± 0.006	0.69± 0.007	0.40 ± 0.005
	Information gain	0.13 ± 0.007	0.11 ± 0.006	0.98± 0.002	0.11 ± 0.006
	Random	0.01 ± 0.002	0.01 ± 0.002	0.99± 0.001	0.003± 0.002
Ref.	**Odds ratio**	**0.53** ± 0.006	**0.64** ± 0.006	**0.51**± 0.007	**0.81** ± 0.008
	Cross entropy Txt	0.52 ± 0.008	**0.60** ± 0.010	**0.62**± 0.003	**0.71** ± 0.010
	Mutual information Txt	0.52 ± 0.010	0.55 ± 0.010	0.73± 0.010	0.60 ± 0.020
	Term frequency	0.50 ± 0.010	0.53 ± 0.010	0.78± 0.005	0.57 ± 0.010
	Weight of evidence Txt	0.20 ± 0.008	0.22 ± 0.010	0.80± 0.009	0.31 ± 0.010
	Information gain	0.25 ± 0.007	0.22 ± 0.007	0.99± 0.002	0.21 ± 0.006
	Random	0.07 ± 0.006	0.06 ± 0.005	0.99± 0.001	0.05 ± 0.005

Table 2: Comparison of feature scoring measures for the problem of keyword prediction on five domains formed from the Yahoo hierarchy. For each domain, the compared feature scoring measures are sorted according to their performance in F2-measure. The values of F1-measure, Precision and Recall are given for better understanding. We give averages with standard errors calculated over 5 runs.

call per document calculated for the fixed probability threshold (experimentally set to 0.95 (Grobelnik and Mladenić, 1998)). Precision can be seen as the classification accuracy calculated only for positive examples, while Recall is the proportion of positive examples the system recognized as positive (values in [0..1]). If testing example is originally assigned to several categories, all these categories are taken as correct and compared to the set of predicted categories. We perform this comparison in two ways: (1) by taking into account proximity to the correct category using keywords assigned to each category (keyword prediction) and (2) by counting only prediction of the correct category (category prediction).

Additional to Precision and Recall we report F_2-measure that is a combination of the two, commonly used when we care more about Recall than about Precision. Tables 2 and 3 give results of the comparison (1) and (2) respectively. We observe performance (F2-measure) for the best performing number of selected features that is in most cases vector size 1 (meaning select as many features as there are features that occur in positive examples). To get an idea about the actual number of the used features, vector size 1 means in average over categories: on *'Entertainment'* 58 out of 30,998 features (0.2%), on *'Arts and Humanities'* 65 out of 11,473 features (0.7%), on *'Computers and Internet'* 42 out of 7,631 features (0.62%), on *'Education'*

Dom. name	Scoring measure	Average on category prediction			
		F1-measure	F2-measure	Precision	Recall
Ent.	**Odds ratio**	**0.29**± 0.002	**0.30** ± 0.003	**0.41** ± 0.004	**0.34** ± 0.003
	Term frequency	0.24± 0.003	0.27 ± 0.003	0.38 ± 0.003	0.34 ± 0.003
	Mutual information Txt	0.22± 0.004	0.23 ± 0.004	0.57 ± 0.006	0.29 ± 0.007
	Information gain	0.25± 0.002	0.20 ± 0.003	0.87 ± 0.002	0.17 ± 0.002
	Cross entropy Txt	0.15± 0.002	0.18 ± 0.002	0.29 ± 0.005	0.28 ± 0.005
	Weight of evidence Txt	0.14 ± 0.003	0.11 ± 0.002	0.67 ± 0.005	0.10 ± 0.003
	Random	0.001± 0.0001	0.001± 0.0002	0.99 ± 0.007	0.001±0.0002
Arts.	**Odds ratio**	**0.29**± 0.002	**0.32** ± 0.004	**0.36** ± 0.005	**0.38** ± 0.004
	Term frequency	0.28± 0.002	0.29 ± 0.003	0.43 ± 0.004	0.34 ± 0.003
	Mutual information Txt	0.24± 0.005	0.25 ± 0.005	0.56 ± 0.006	0.31 ± 0.007
	Cross entropy Txt	0.18± 0.002	0.22 ± 0.003	0.27 ± 0.005	0.32 ± 0.006
	Information gain	0.21± 0.002	0.17 ± 0.002	0.93 ± 0.003	0.15 ± 0.002
	Random	0.0012± 0.0001	0.001± 0.0003	0.99 ± 0.006	0.001±0.0002
	Weight of evidence Txt	0.11 ± 0.003	0.09 ± 0.002	0.55 ± 0.004	0.079± 0.002
Comp.	**Odds ratio**	**0.30**± 0.002	**0.33** ± 0.002	**0.36** ± 0.009	**0.57** ± 0.005
	Term frequency	0.27± 0.003	0.26 ± 0.002	0.45 ± 0.003	0.27 ± 0.003
	Mutual information Txt	0.25± 0.003	0.24 ± 0.004	0.60 ± 0.006	0.26 ± 0.006
	Cross entropy Txt	0.19± 0.005	0.21 ± 0.004	0.28 ± 0.004	0.27 ± 0.002
	Information gain	0.19± 0.007	0.14 ± 0.006	0.94 ± 0.004	0.12 ± 0.005
	Weight of evidence Txt	0.09± 0.002	0.07 ± 0.002	0.59 ± 0.010	0.06 ± 0.001
	Random	0.001± 0.0003	0.001± 0.0002	0.99 ± 0.001	0.001± 0.0002
Edu.	**Mutual information Txt**	**0.40**± 0.006	**0.45** ± 0.009	**0.52** ± 0.010	**0.53** ± 0.010
	Odds ratio	0.32± 0.005	**0.43** ± 0.005	**0.36** ± 0.009	**0.57** ± 0.010
	Term frequency	**0.40**± 0.010	**0.42** ± 0.010	**0.57** ± 0.010	**0.45** ± 0.020
	Weight of evidence Txt	0.05± 0.006	**0.42** ± 0.005	**0.53** ± 0.020	**0.04** ± 0.005
	Cross entropy Txt	0.2 ± 0.006	0.26 ± 0.004	0.23 ± 0.010	0.37 ± 0.005
	Information gain	0.09± 0.003	0.07 ± 0.002	0.97 ± 0.003	0.07 ± 0.002
	Random	0.01± 0.001	0 ± 0.001	0.99 ± 0.002	0.001± 0.001
Ref.	**Odds ratio**	**0.37**± 0.007	**0.42** ± 0.009	**0.46** ± 0.009	**0.51** ± 0.012
	Mutual information Txt	0.34± 0.005	0.32 ± 0.005	0.69 ± 0.015	0.32 ± 0.006
	Term frequency	0.28± 0.007	0.26 ± 0.070	0.72 ± 0.007	0.26 ± 0.007
	Cross entropy Txt	0.23± 0.005	0.22 ± 0.005	0.50 ± 0.003	0.23 ± 0.005
	Information gain	0.20± 0.003	0.16 ± 0.002	0.99 ± 0.002	0.14 ± 0.002
	Weight of evidence Txt	0.06± 0.007	0.05 ± 0.006	0.70 ± 0.011	0.05 ± 0.004
	Random	0.05± 0.006	0.04 ± 0.005	0.99 ± 0.001	0.04 ± 0.004

Table 3: Comparison of feature scoring measures for the problem of category prediction on five domains formed from the Yahoo hierarchy. For each domain, the compared feature scoring measures are sorted according to their performance in F2-measure. The values of F1-measure, Precision and Recall are given for better understanding. We give averages with standard errors calculated over 5 runs.

85 out of 3,198 features (2.7%), on *'References'* 49 out of 928 features (5.3%). Additionally to the value of F_2-measure, we give values of Precision, Recall and F_1-measure. F_1-measure is included to show how the same feature scoring measures would compare in case we would have a problem where Precision and Recall are equally important. As we can see from Tables 2 and 3, there would not be much difference, Odds ratio and Term frequency would remain the best.

On all domains Odds ratio is among the best performing measures (see Tables 2 and 3) and the best performance is achieved when only a small number of features is used. For instance, in Table 2 on *'Computers and Internet'* Odds ratio achieves F_2-measure of 0.60, Precision of 0.40 and Recall of 0.84, meaning that 40 % of document predicted positive are positive and that 84% of all positive documents are identified. This is consistent with the results reported in text-learning on the problem of predicting clicked hyperlinks from the set of visited Web documents (Mladenić, 1998b) where the feature selection based on Odds ratio achieved the best results. Similar observation regarding the number of features is reported on text categorization in (Yang and Pedersen, 1997), where the reduction of up to 90% in the number of features resulted in either an improvement or no loss in the system performance. Observation of standard error on all five domains confirms that the best performing measures are significantly better

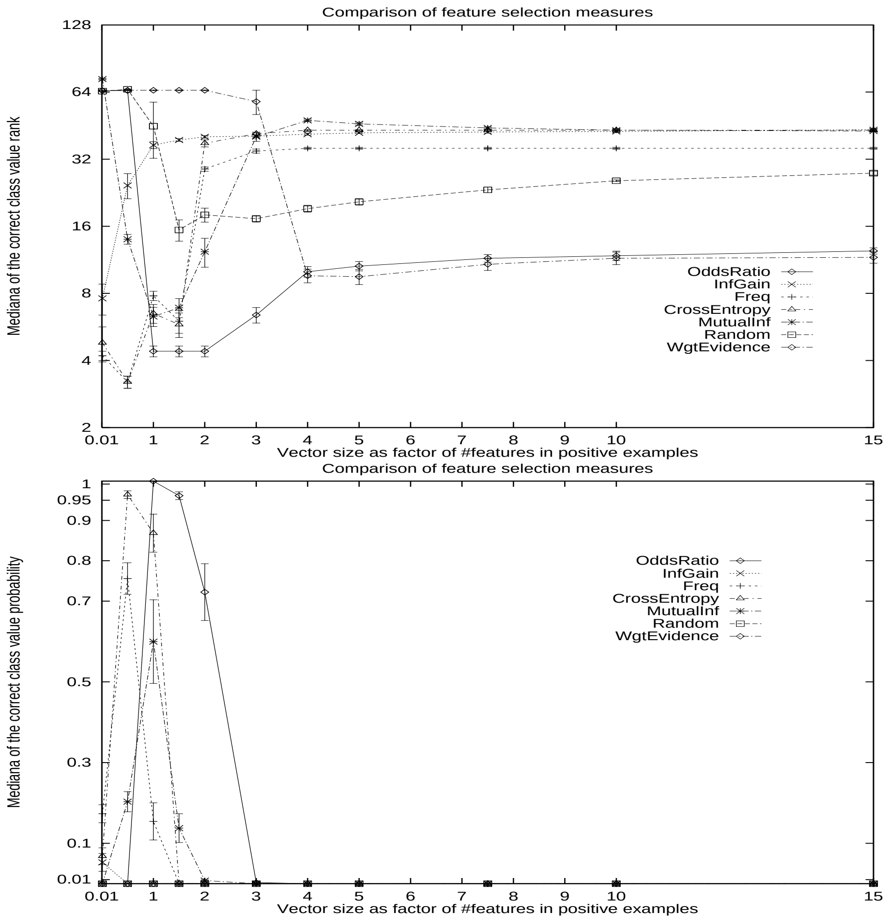

Figure 1: Comparison of (upper) the correct category rank (the lower the better) and (lower) probability for different feature scoring measures on domain *'Reference'*.

than the other tested measures.

Additionally, we report two non-standard but intuitive measures: rank and probability assigned to the correct category. For each testing example we observe a list of categories each assigned a probability as a result of consulting the corresponding subproblem classifier. Sorting categories according to that probability gives ranking that we use to get the rank of the correct category. If there are more correct categories, the one with the highest predicted probability is considered. To get summary results over the testing examples we give median rather than mean, since some of the testing examples are rather non-typical of their category, containing eg., a welcome page or only one sentence asking for language preference or an error message or a page giving redirection.

Rank and probability show that Odds ratio is again the best or one of the two best performing measures. Cross entropy for text and Term frequency achieved similar results as Odds ratio on two out of the five domains. Weight of evidence for text achieved on all domains good rank but low probability, while Information gain performed similar as Random. Tables with results of rank and probability are not included because of the space restrictions. They can be found in (Mladenić 1998). Standard errors confirm that the results are significant. Observation of the average number of considered categories during the classification shows that Odds ratio is considering about 3 times less categories than Cross entropy for text and about 2 times less categories than Term Frequency.

For illustration of the influence of the number of selected features we show graphs for domain *'Reference'*. Figure 1 gives median for the correct category rank and probability on domain *'Reference'*. It can be seen

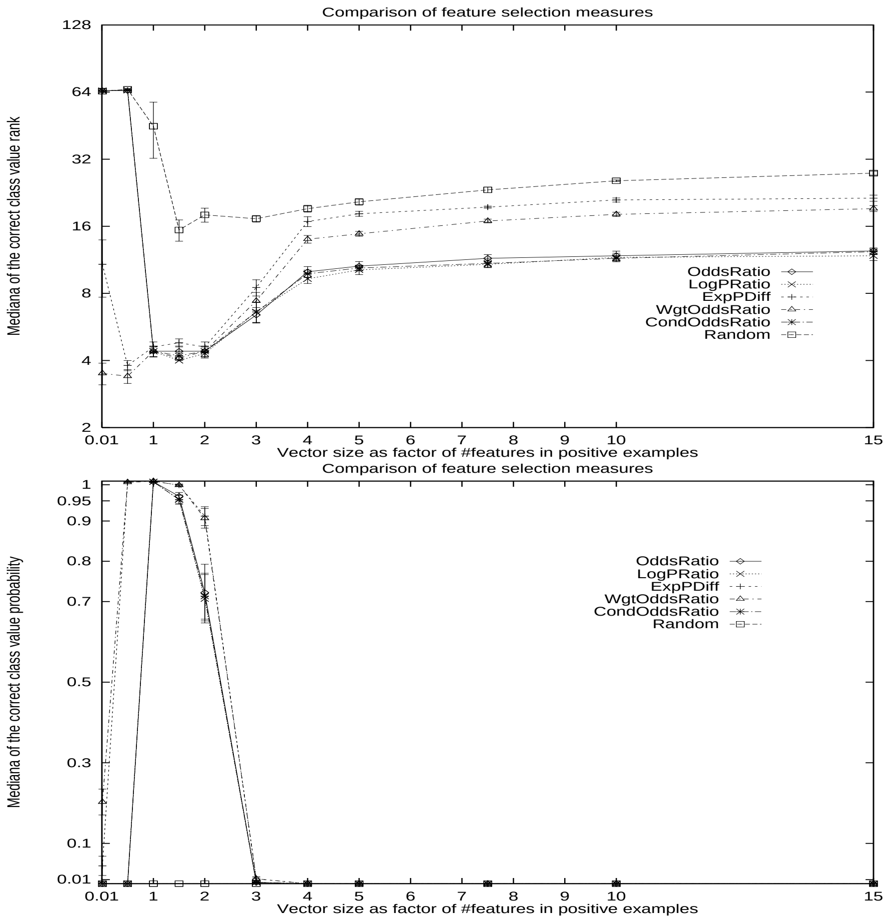

Figure 2: Comparison of Odds ratio and its variants on domain *'References'* defined from the Yahoo hierarchy. We give median of the correct category rank (left) and probability (right).

that the best performance in rank and probability is achieved by Odds ratio and Cross entropy for text using relatively small number of features (vector size 0.5-1.5). For instance, on *'References'* the mediana of the correct category rank is 3.8 the mediana of the correct category probability over 0.99, ie. the half of the testing examples are assigned rank up to 3.8 and probability > 0.99 for the correct category. For larger feature subsets the rank became insensitive to the additional features, while the probability is almost 0.

To show the efficiency of the used scoring measure we give in Figure 3 the influence of the relative number of selected features to the number of categories considered in classification. For Random and Information gain this number grows with the growing number of selected features (vector size). For the other measures, the number of considered categories is mostly stable when using more features (eg., using $\geq$ 50 features for domain *'Reference'*). The lowest number of categories is considered by Weight of evidence for text, but this measure did not achieve very good classification results. Odds ratio is the second best in the low number of considered categories and this is also one of the best performing measures. Cross entropy for text that is also among the best performing measures is the worst in the number of considered categories.

There is no significant difference in the performance between Odds ratio and any of its four variants we tested (for illustration, see Figure 2). This shows that the most important characteristics of Odds ratio are included in its variants. It also shows that we didn't get any significant improvement by including probability of word occurrence (Weighted Odds ratio) nor by including features characteristic for negative examples (Conditional Odds ratio).

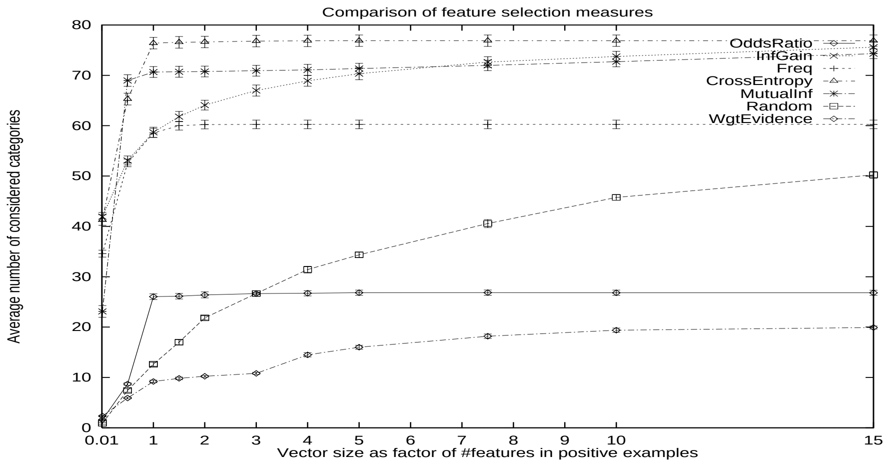

Figure 3: Comparison of the number of considered categories (the lower the better) for different feature scoring measures on domain *'Reference'*.

5 Discussion

In our experiments the best performing feature scoring measures are Odds ratio and its variants, while the worst are Random and Information gain. The other perform comparable or worse than Odds ratio and better than Information gain. A closer look to the highly scored features by Odds ratio and by Information gain explains the huge difference in their performance and poor performance of Information gain. It also indicates how important is to consider domain and algorithm characteristics.

5.1 Unbalanced class and feature value distribution

Yang and Pedersen (Yang and Pedersen, 1997) give experimental comparison of five measures for feature selection in text categorization on word-vectors representing documents. Their experiments confirm our observation that a rather small feature subset should be used since it gives either better or as good results as large feature subset or all the features. We also agree in the observation that a simple frequency of a feature (used after stop words removal and calculated either using term or document frequency) achieves very good results. Our results disagree in the performance of Information gain. Information gain was one of the best performing measures on problems addressed by Yang and Pedersen (Yang and Pedersen, 1997), while we found it performing similar as random on our domains. The reason for that we find in the difference in our domains and classification algorithms. We both use learning for document categorization but different approach to domain definition is used. Their domain is defined to include one class value for each category, while we split the problem into subproblems each corresponding to one category and having binary-valued class. The result of learning is in our case a set of specialized classifiers instead of one huge classifier including the union of features characteristic for different categories. The other important issue is the used classification algorithm. Yang and Pedersen (Yang an Pedersen, 1997) used k-Nearest Neighbor and Linear Least Square Fit mapping. The specific of the Naive Bayesian classifier we are using is that it considers only features that occur in a classification document. This means that highly scored features should be features that will probably occur in new documents.

From the formula for Information gain it can be seen that no difference between the class values or between the feature values is considered. Here we have unbalanced class distribution and also highly unbalanced feature value distribution. When calculating feature score we observe two values for each feature (word sequence): occurs or does not occur in a document. The prior probability that a word sequence occurs in a document $P(W)$ is rather small. Most of the features selected by Information gain are features with the majority feature value ($P(\overline{W})$ is high). If $P(\overline{W}) >> P(W)$ then the high value of Information gain in most cases means that the second part $P(\overline{W})\sum_i P(C_i|\overline{W})\log\frac{P(C_i|\overline{W})}{P(C_i)}$ of the Information gain formula is high. In other words, knowing that W does not occur in a document brings useful

Measure name	target class value	common features	using class value	using feature absence	perf.
Odds ratio	**Yes**	No	Yes	No	great
Weighted Odds ratio	**Yes**	**Yes**	Yes	No	great
Log. prob. ratio	**Yes**	No	Yes	No	great
Exp. prob. difference	**Yes**	No	Yes	No	great
Conditional Odds ratio	**Yes**	No	Yes	No	great
Cross entropy for text	No	**Yes**	Yes	**No**	good
Term frequency	No	**Yes**	No	No	good
Weight of evidence for text	No	Yes	Yes	No	mild
Mutual information for text	No	*No*	Yes	No	mild
Information gain	No	Yes	Yes	*Yes*	poor
Random	No	No	No	No	poor

Table 4: Comparison of feature scoring measures by their characteristics and performance (perf.) on our domains.

information about the class value. Intuitively, when classifying a new document, a better classification results are expected if the classification is based on words that occur in a document. It is possible that absence of some words in a document is very informative and this is taken into account by the new feature scoring measure we named Conditional Odds ratio. The problem is that the classification based mostly on the absence of words is usually harder and requires larger feature subset than the classification based on word occurrences. Cross entropy for text makes distinction between the feature values and achieves good results in our experiments. It equals Information gain calculated only for one of the feature values - W meaning word does occur. Odds ratio and its variants achieve better results than Cross entropy for text. They also ignore feature absence but additionally take into account that there is a target class value. While Odds ratio and its variants favor features characteristic for positive examples (high $P(W|\text{'}pos\text{'})$), the other tested measures make no distinction between the class values. Moreover, Information gain makes no distinction between the feature values (it is using feature absence). Since we have unbalanced class distribution with over 90% of examples having negative class value, most of the features are characteristic for negative examples. This means that most of the features highly scored by Information gain are either informative when they do not occur in a document or they are characteristic for negative class value (just the opposite of Odds ratio!).

In (Yang and Pedersen, 1997) the tested scoring measures are compared across several criteria: favoring common features, using class value, using feature absence and experimental performance. They conclude that the three measures (Frequency, Information gain and χ^2 statistics) achieving the best results on their domains (Reuters-22173, a subset of MEDLINE) all favor common features.

5.2 Conclusions

Table 4 gives characterization of the feature scoring measures compared in our experiments. We also observe whether the feature scoring measure makes difference between the class values, since this is important for our domains having one of the class values as the target class value. The best results are achieved by Odds ratio and its variants that assume the problem has binary-valued class and one of the class values is the target class value (asymmetric misclassification costs). Between these measures no significant difference in performance was observed. The next group of measures achieving good results all favor common features (Cross entropy for text, Term frequency, Weight of evidence for text). Mutual information for text differs from Cross entropy for text only in not favoring frequent features and achieves worse results. Information gain differs from Cross entropy for text only in using feature absence as well as feature presence and achieves poor results.

Our conclusion is that in general the most important characteristics of a good feature scoring measure for text are: favoring common features and considering domain and algorithm characteristics. For learning algorithms that make difference between feature presence and absence, such as the Naive Bayesian classifier used here, it is important that a scoring measure also makes this difference. For the domains with binary-valued class where one class value is the target class value, the most important characteristics of a good feature scoring measure is to make difference between the class values and favor features characteristic for the target class value. In this case, favoring common features is not an important issue. Namely, Weighted odds ratio that favors common features is not performing better than Odds ratio.

Instead of using a filtering approach to feature selection that ignores the learning algorithm or using the wrapper approach that uses the learning algorithm as a 'black-box', we suggest that domain and algorithm characteristics are studied in advance. We applied the Naive Bayesian classifier to the domains that have an unbalanced class and feature value distribution and asymmetric misclassification costs, where the minority class value is the target class value. Experimental comparison of different feature scoring measures used in feature selection shows that Odds ratio (and its variants) achieve the best results. Our classifier uses the same conditional probability as used in Odds ratio for scoring the features. In this way, the selected features are features expected to have the greatest influence to the posterior probability of class values returned by the Naive Bayesian classifier.

More precisely, let us consider domain and algorithm characteristics to see which features should be selected. In our case the majority class value is negative ($P(neg) > P(pos)$). If we want to identify the positive documents then the Naive Bayesian classifier should return $P(pos|Doc) > P(neg|Doc)$ for positive documents. This can be achieved only if $\Pi_{W_j \in Doc} P(W_j|pos)^{TF(W_j,Doc)} > \Pi_{W_j \in Doc} P(W_j|neg)^{TF(W_j,Doc)}$. At the same time, Odds ratio favors words that have $P(W_j|pos) > P(W_j|neg)$. Having many such words selected for learning means, that we have good chances to get the above product in the classifier higher for the positive than for the negative class value. Moreover, our experiments suggest that we should select as many best features as there are features (word sequences) that occur in the positive examples. Closer look to the features sorted according to Odds ratio show that this approximately means simply select all the features that occur in positive examples without performing any feature scoring. In general, we can conclude that on such problems with unbalanced class distribution and asymmetric misclassification costs, features characteristic for the positive examples should be selected.

References

Filo, D., Yang, J. (1997), Yahoo! Inc.

Grobelnik, M. & Mladenić, D. (1998), Efficient text categorization, *ECML-98 Workshop on Text Mining.*

John, G.H., Kohavi, R., Pfleger, K. (1994), Irrelevant Features and the Subset Selection Problem, *Proc. of the 11th International Conference on Machine Learning ICML94*, pp.121-129.

Koller, D., Sahami, M. (1997), Hierarchically classifying documents using very few words, *Proc. of the 14th International Conference on Machine Learning.*

Kononenko, I. (1995), On biases estimating multivalued attributes. *Proc. of the 14th International Joint Conference on Artificial Intelligence IJCAI-95.*

Lewis, D.D. (1995), Evaluating and optimizating autonomous text classification systems. *Proc. of the 18th Annual International ACM-SIGIR Conference on Recsearch and Development in Information Retrieval.*

McCallum, A., Nigam, K. (1998), A comparison of event models for naive Bayes text classifiers. *AAAI-98 Workshop on Learning for Text Categorization.*

Mladenić, D. (1998), Machine Learning on non-homogeneous, distributed text data, *PhD thesis*, University of Ljubljana, Slovenia, October, 1998.
http://www.cs.cmu.edu/~TextLearning/pww/PhD.html

Mladenić, D. (1998a), Turning Yahoo into an Automatic Web-Page Classifier. *Proc. of the 13th European Conference on Aritficial Intelligence ECAI'98.*

Mladenić, D. (1998b), Feature subset selection in text-learning, *Proc. of the 10th European Conference on Machine Learning ECML98.*

Mladenić, D., & Grobelnik, M. (1998), Word sequences as features in text-learning. *Proc. of the Seventh Electrotechnical and Computer Sc. Conference ERK'98*, pp.145-148, Slovenia: IEEE section.

Quinlan, J.R. (1993), Constructing Decision Tree in *C4.5: Programs for Machine Learning*, pp.17-26, Morgan Kaufman Publishers.

Pazzani, M., & Billsus, D. (1997), Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning, 27*, pp.313-331.

van Rijsbergen, C.J,. Harper, D.J., Porter, M.F. (1981), The selection of good search terms, *Information Processing & Management*, 17, pp.77-91.

Yang, Y., Pedersen, J.O. (1997), A Comparative Study on Feature Selection in Text Categorization, *Proc. of the 14th International Conference on Machine Learning ICML97*, pp.412-420.