A Framework for Characterizing Feature Weighting and Selection Methods in Text Classification

Janez Brank
Natasa Milic-Frayling

January 31, 2005

MSR-TR-2005-09

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

A Framework for Characterizing Feature Weighting and Selection Methods in Text Classification

Janez Brank
Jozef Stefan Institute
Ljubljana, Slovenia
Janez.Brank@ijs.si

Natasa Milic-Frayling
Microsoft Research Ltd.
Cambridge, UK
natasamf@microsoft.com

ABSTRACT

Optimizing performance of classification models often involves feature selection to eliminate noise from the feature set or reduce computational complexity by controlling the dimensionality of the feature space. A refinement of the feature set is typically performed in two steps: by scoring and ranking the features and then applying a selection criterion. Empirical studies that explore the effectiveness of feature selection methods are typically limited to identifying the number or percentage of features to be retained in order to maximize the classification performance. Since no further characterizations of the feature set are considered beyond its size, we currently have a limited understanding of the relationship between the classifier performance and the properties of the selected set of features. This paper presents a framework for characterizing feature weighting methods and selected features sets and exploring how these characteristics account for the performance of a given classifier. We illustrate the use of two feature set statistics: cumulative information gain of the ranked features and the sparsity of data representation that results from the selected feature set. We apply a novel approach of synthesizing ranked lists of features that satisfy given cumulative information gain and sparsity constraints. We show how the use of synthesized rankings enables us to investigate the degree to which the feature set properties explain the behaviour of a classifier, e.g., Naïve Bayes classifier, when used in conjunction with different feature weighting schemes.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: *retrieval models, selection process.* I.2.6 [**Learning**]: *Concept learning*.

Keywords

Feature weighting, feature selection, text categorization, information retrieval, vector sparsity, density

1. INTRODUCTION

In text classification, feature selection is widely used to control the dimensionality of the feature space or reduce noise in the feature set. That procedure typically involves two steps: first applying a feature weighing scheme to score and rank the features and then specifying the feature selection criterion that determines which portion of the ranked list of features should be retained for further processing. In most cases the selection criterion is the top N or top N percent of the ranked features. Identifying N that optimizes the performance of the classifier is treated as an empirical question that requires training and evaluation of the classification models for a range of values for N. As a result, no systematic investigation of the feature set properties has been attempted to link the characteristics of the selected feature sets and the classification performance.

This paper introduces a framework that fulfills this role and, to our knowledge, is the first of that kind. It involves defining a single or multiple functions $N \rightarrow F(N)$ for the ranked list of features, which capture properties of the feature sets that we wish to observe. An example of such a property is the cumulative information gain of features included in the selected feature set: one can observe the change in value of the cumulative information gain $I(N)$ as the feature set grows with the increased N. This is represented as a qualitative curve $N \rightarrow I(N)$ associated with the particular feature ranking.

On the other hand, for a given weighting scheme and the corresponding feature ranking we observe the performance statistics of a given classifier as a function $N \rightarrow P(N)$ of the rank N, which defines the set of features and the its size. The questions then arise:

- To which degree the particular property of the feature set, captured by $F(N)$, characterizes the given term weighting scheme. In other words, how close a randomly generated feature ranking that satisfies the characteristic function $N \rightarrow F(N)$ resembles the original feature ranking which resulted from the given term weighting scheme?
- To which degree the deviation from the original ranking influences the performance of the classifier, i.e., if some other ranking satisfies the constraint $N \rightarrow F(N)$, how close does it reproduce the original performance curve $N \rightarrow P(N)$?

We illustrate the use of this framework to investigate the performance of the Naïve Bayes classifier when used in conjunction with five feature weighing schemes. As examples of the *feature set characteristic functions* $N \rightarrow F(N)$ we consider the *cumulative information gain* and the *sparsity* of the document vectors induced by the selected feature set.

In the following section we describe general issues associated with feature selection methods and further discuss the proposed framework. We briefly introduce the feature weighting methods used in our experiments and present the experiment design and findings. We refer to the related work throughout the paper, as appropriate. We conclude with the summary of our work and a set of open questions that will be addressed in our future research.

2. FEATURE SELECTION RESEARCH

Research in feature selection methods has been to a large extent focused on investigating how effective a given feature weighting method is in combination with a particular classification method. This typically leads to a more or less systematic evaluation of the classification performance for a range of feature set sizes, determined by different cut-off levels of the feature ranked list [6][1][9][13]. Our objective is to move that research forward by investigating the dependency of the classifier performance not only on the feature set size but the other properties of the selected feature sets.

For the purpose of this discussion we equate the feature weighting algorithm with the feature ranking that the assigned weights induce. We fully realize that in some instances the ranking of the features is not uniquely defined by assigned the weights since multiple features may be given the same value. Furthermore, while the weighing scheme cannot differentiate between different permutations of features with same score, their ordering and selection may have a significant impact on the classifier performance. However, for the sake of clarity and simplicity we disregard that issue for now and focus on gaining a better understanding of the feature selection methods and the resulting feature sets.

2.1 Combining Feature Weighting and Classification Models

Studies have shown that different feature ranking methods may lead to very different classification performance for the same classification model [13]. This can be observed by considering how the performance depends on the feature set size N, i.e., the performance curves $N \rightarrow P(N)$, where $P(N)$ is some suitable performance measure (in our experiments, we used the well-known F_1-measure). Figure 3 shows the performance of the Naïve Bayes algorithm when five different feature ranking methods are used (see Section 3). Consequently, setting in advance a fixed number of features to be retained from the ranked list and used for classification may lead to a dramatically different performance, depending on the feature weighting scheme.

An intuitive explanation of this observation is that a feature weighting can be more or less coordinated with the classification model, in the sense that they may be governed by the same or distinct theoretical models. In that respect, feature scoring using Odds Ratio is seen as a good match for the Naive Bayes classifier and has been shown to improve its performance [3]. However, this argumentation is not satisfactory, in particular, when the feature ranking is based on a scoring scheme that cannot be expressed in a clear analytical form, such as scoring features based on the weights generated by the normal of the linear classifiers, such as SVM and Perceptron [10][13]. Our position is that the key for understanding the interaction between the feature selection and the classification methods lies in connecting statistical properties of the selected feature sets and the classifier performance. Thus we propose a framework that enables us to conduct that investigation.

It is important to differentiate between statistics used to score and rank individual features by the weighting scheme and the statistical properties of the selected feature set. In most cases the classification model does not explicitly incorporate the weights assigned to the features during ranking. Thus, the performance of the classifier is a function of the ranking itself and the statistical properties of the selected feature set that is used by the classification method.

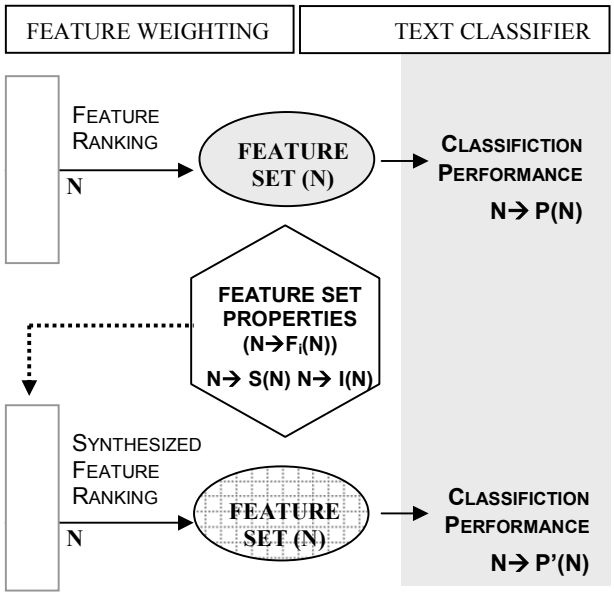

Figure 1. Framework. Comparison of the classifier performance for the synthesized feature ranking – which satisfies the same feature set properties as the original ranking (constraints $N \rightarrow F_i(N)$), shows how effective these properties are in explaining the classifier performance.

3. A FRAMEWORK FOR CHARACTERIZING FEATURE RANKINGS

We propose to investigate the dependency of the text classifier on the properties of the feature set through the framework that involves the following concepts and procedures:

- *Ranked feature set*. For a given feature set, consider a weighting scheme. The result is a ranked set of features.
- *Statistical profile of the feature set.* Characterize the set of ranked features by relevant statistics F, i.e., generate a statistical profile by observing one or several characteristic functions and the corresponding characteristic curve $N \rightarrow F(N)$.
- *Performance statistics for the feature set.* For the same rank list, consider the performance curve of the classifier, showing the change of performance statistics (e.g. the F_1-measure) with the size N of the used feature set.
- *Synthesized ranked sets.* Assess to what degree the statistical profile of the feature set explains the feature weighing function, i.e., the resulting feature ranking. We propose to accomplish that in the following way. We treat the statistical profile – i.e., characteristic curves of the feature ranking, as constraints. We then generate a ranked list of features that satisfies the constraints, by (randomly) selecting terms from the entire feature set and adding them to the ranked list. The result is a synthesized ranking for which the characteristic functions are approximately the same as those of the original

ranking. We can investigate the discrepancy between the original and synthesized ranking. For example, we can look at the intersection of the feature sets when the cut-off is set to N for both ranked lists.

- *Classification peformance for the synthesized feature ranking.* We look at the classification performance for the synthesized set. If we observe degradation in the performance of the classifier we can conclude that the observed characteristic function, or constraint, only weakly characterizes the classifier performance. Since we can create synthesized sets that correspond to any number of constraints we can observe the dependency of the classifier performance on any number of constraints, generating the appropriate feature ranking. Those combinations of constraints that yield higher performance are considered better in explaining the behavior of the classification performance.

The underlying assumptions of this approach are:

- The performance of a given feature ranking (when plugged into a given classification method) can be reasonably explained by a finite number of statistical properties related to the used feature set.
- Given a feature ranking, it can be characterized (modulo permutation of the features with similar score in the ranking) by a sufficient number of constraints $N \rightarrow F_i(N)$, expressed in the form of characteristic curves.

3.1 Definition of the Characteristic Functions

As an illustration of the framework application, we here consider text classification using the Naïve Bayes classifier and two feature set statistics: the *sparsity* and the *cumulative information gain*. They induce two characteristic functions for the feature sets and serve as constraints for generating synthesized feature rankings.

3.1.1 Sparsity

It has been observed that some feature weighting schemes rank highly those features that have low distribution across the data corpus and thus yield very sparse representation of documents [10]. More precisely, if we retain only a small number N of top ranked features, the average number of non-zero components per document vector is low. Thus, a significant number of documents is represented as zero vectors and do not participate in the classifier training. For that reason, a useful characteristic of a feature set is the *vector sparsity statistics* (or *vector density*) calculated as the average number of non-zero components across document vectors. Applied to the ranked feature list, this leads to a *sparsity curve* $N \rightarrow S(N)$ associated with a particular feature scoring scheme. Figure 1 shows sparsity curves for several feature scoring schemes (see Section 4.1).

As sparsity is closely related to the *distribution of features* in the corpus and its inverse, known as *idf* or *feature specificity* or *rarity*, the sparsity curves implicitly show how strong the influence of these statistics is on the feature scores. This is particularly useful when feature weighting is not based on an explicit analytic formula but, for example, obtained from linear classifiers such as Support Vector Machines (SVM) [10].

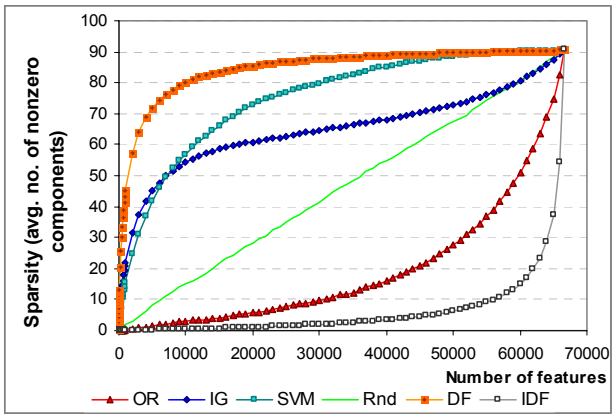

Figure 2. Sparsity curves for five feature ranking schemes: Odds Ratio (OR), Information Gain (IG), Support Vector Machines (SVM), Document Frequency (DF), Inverse Document Frequency (IDF) and a random ranking of features (Rnd).

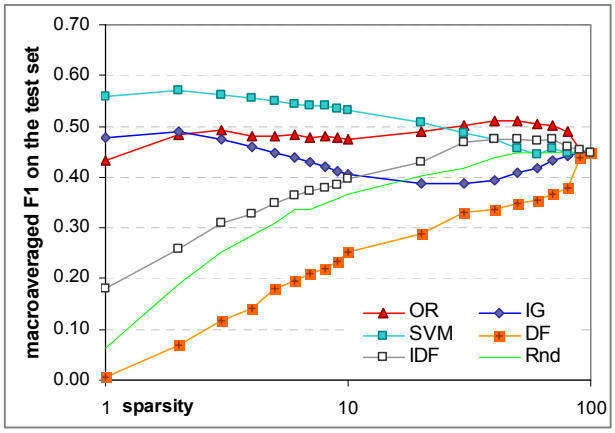

Figure 3. Macroaveraged F_1 measure for the Naïve Bayes classifier using different feature ranking schemes.

For the sake of clarity, when we plot the performance curves for different ranking schemes we use the sparsity as the independent variable instead of the feature set size N. For example, Odds Ratio tends to rank highly many rare features, as long as they don't appear in any negative documents. Therefore, it needs a considerable number of features to achieve a non-zero sparsity of vectors and yield a good classification performance. Table 1 and 2 show the relationship between the predefined sparsity levels and the corresponding number of features N for different feature ranking schemes.

3.1.2 Cumulative Information Gain

Furthermore, since we are considering feature selection in the context of text classification, it is natural to expect that one of the important aspects of each feature set is the distribution of constituent features among documents associated with each class (positive and negative). One measure that combines evidence from the positive and negative class is *information gain*. We thus use the *cumulative information gain* to characterize the set of selected features and consider the *cumulative information gain curves*

curves $N \rightarrow I(N)$ for each of the ranking functions. $I(N)$ is defined as the sum of the information gain scores $IG(class, feature)$ for the N top-ranking features.

The justification for the use of I(N) can be made also made from the perspective of uncertainty that remains regarding the class label of a document if we know the values of e.g. N top-ranking features. If the feature ranking puts features that are informative of the class label high in the rank, the uncertainty will be low. It could be measured using conditional entropy $H(C|T_1,\ldots,T_N)$ where C is the class label and $T_1,\ldots,T_N$ are the top-ranking N features. In practice, it would be difficult to accurately compute this entropy for large values of N, or to estimate the underlying probability distribution. However, if we assume that the terms are statistically independent, the conditional entropy can be expressed as

$$\begin{aligned} H(C|T_1,\ldots,T_N) &= H(C,T_1,\ldots,T_N) - H(T_1,\ldots,T_N) = \\ &= H(C) + H(T_1,\ldots,T_N|C) - H(T_1,\ldots,T_N) = \\ &= H(C) + \Sigma_{i=1,\ldots,N} H(T_i|C) - \Sigma_{i=1,\ldots,N} H(T_i) \\ &= H(C) - \Sigma_{i=1,\ldots,N} [H(C)+H(T_i)-H(T_i,C)] \\ &= H(C) - \Sigma_{i=1,\ldots,N} MI(C;T_i). \end{aligned}$$

Here MI denotes the mutual information of C and T_i. This is also known as information gain, expressing the amount of information gained about one variable if we learn the value of the other variable. The sum $\Sigma_{i=1,\ldots,N} MI(C;T_i)$ is exactly the cumulative information gain which we use to characterize feature rankings. Thus, the cumulative information gain is a simplification of the formula for the amount of information about the class that is contained in the first N top-ranking features; the simplification would be accurate if the features were independent variables.

3.2 Synthesis of Feature Rankings with Constraints

We can use a pair of characteristic curves (a sparsity curve $S(N)$ and a cumulative information gain curve $I(N)$) as constraints to generate a new, synthesized feature ranking in such a way that its sparsity curve and cumulative IG curve are approximately the same as the original curves. The algorithm that se currently use for generating such synthetic feature rankings is as follows:

$n_1 := 1$; $F :=$ {set of all features};
while $n_1 \le$ the total number of features do
  $n_2 := n_1 + 1$;
  while $n_2 \le$ the total number of features and $n_2 - n_1 < \theta_n$
    and $S(n_2)-S(n_1) < \theta_S$ and $I(n_2)-I(n_1) < \theta_I$ do $n_2 := n_1 + 1$;
  target average DF := $(S(n_2) - S(n_1))/(n_2 - n_1)$;
  target average IG := $(I(n_2) - I(n_1))/(n_2 - n_1)$;
  select $n_2 - n_1$ features from F such that their average DF
    and average IR approximately match the target values
    computed above;
  append these features to the ranking which we are
    synthesizing, and remove them from F;
  $n_1 := n_2$;
end while;

The thresholds θ_n, θ_S, and θ_I define how fine-grained the approximation of the target characteristic curves (constraints) is meant to be. Small thresholds may result in a synthesized ranking which is closer to the original one. However, that also implies that the target characteristic curves are known in great detail, which may not be the case in practice. Currently our thresholds are set at 1% of the total number of features (for θ_n), 1% of the total sparsity (for θ_S), and 1% of the total cumulative information gain (for θ_I).

Our implementation uses a quad-tree to index the set F of features not yet included in the ranking under construction; each feature is represented by a ⟨document frequency(DF), information gain(IG)⟩ pair. To select the next batch of n_2-n_1 features, we perform a spatial query in the quad-tree to find features with DF and IG near the desired average values. This is convenient from the point of view of implementation efficiency, but imposes an unnecessary additional constraint; the average DF and IG of the current batch of features should be close to the target values, but the DF and IG of each individual feature need not be close to the target average values. This additional constraint sometimes makes it difficult to synthesize a ranking whose characteristic curves match those of the target ranking. The design of a better synthesis algorithm will be a subject of our future work.

4. EXPERIMENT SET UP

4.1 Feature Weighting Schemes

In text classification, numerous feature weighting methods have been used to assign scores and rank features, including Odds Ratio, Information Gain, χ^2, term strength, weights from a linear classifier [10][13], etc. Even the simple document frequency (DF) has been found to perform well in conjunction with the *k*-Nearest Neighbor method [6][5]. Here we consider a selection of five weighting schemes.

4.1.1 Odds Ratio (OR)

The Odds Ratio score of a term t is calculated as follows:

$$OR = \log[odds(t|positive)/odds(t|negative)]$$

where $odds(t|c) = P(t|c)/(1-P(t|c))$, c denotes a class having two possible values: *positive*, *negative*. $P(t|c)$ is the probability that the term t is present in a randomly chosen document from class c.

This measure gives a high score to features typical of positive documents and a low score to those typical of negative documents. Note that features which occur in very few positive documents can get very high scores as long as they do not occur in negative documents. In this manner rare rather than representative features of positive documents obtain high scores. The method has been used in conjunction with Naive Bayes for categorizing web pages based on profiles of web usage [3] and for classifying Yahoo data into Yahoo categories [5].

4.1.2 Information Gain (IG)

Using the information-theoretic definition of mutual information we define information gain (IG) of a term t as:

$$\begin{aligned} IG(T) &= H(C) - H(C|T) \\ &= \Sigma_{\tau,c} P(C=c,T=\tau) \ln[P(C=c,T=\tau)/P(C=c)P(T=\tau)]. \end{aligned}$$

Here, τ ranges over {*present*, *absent*} and c ranges over {*positive*, *negative*}. In effect, IG is the amount of information about C, the class label, gained by knowing T (i.e. by knowing the presence or absence of a given word).

4.1.3 Feature weights from linear classifiers

Linear classifiers such as e.g. linear SVM [8] calculate predictions in the form: $prediction(\mathbf{x}) = \text{sgn}(\mathbf{w}^T\mathbf{x} + b) = \text{sgn}(\Sigma_j w_j x_j + b)$, where w_j and x_j are vector components that correspond to the

feature i. The vector **w** also represents the *normal* to the hyperplane determined by the classifier, which separates positive from negative class examples. Features whose w_i is close to 0 have a small influence on the predictions; it is therefore reasonable to assume that they are also not very important for learning. The SVM feature ranking thus involves ordering the features based on the absolute value $|w_i|$ of the coefficient they have in the representation of the normal. This approach has been explored in detail for linear SVMs in [10] [11].

Since training an SVM model is relatively expensive, in practice this method is applied to a subset of the training data in order to obtain feature ranking and after the final set of features is selected the SVM classifier is trained on the full feature set [10][13]. In this study we disregard this issue and assume that the SVM normal is based on initial training over the full set of training data.

4.1.4 Distribution based weights

Use of simple *feature distribution* (DF) in the corpus for feature selection has been already explored in a number of comparative studies, pointing its beneficial effect on the classifier performance [6]. For us, DF and the *inverse document frequency* (IDF) are of interest since they are often used in some form by classification methods. They are also most explicitly related to the concept of vector sparsity.

4.2 Naïve Bayes Classification Algorithm

For Naïve Bayes classification we use the multinomial model as described by McCallum and Nigam [12]. The predicted class for document d is the one that maximizes the posterior probability $P(c|d)$, which is proportional to $P(c)\Pi_w P(t|c)^{TF(t,d)}$. Here $P(c)$ is the prior probability that a document belongs to class c, $P(t|c)$ is the probability that a word w, chosen randomly in a document from class c equals t, and $TF(t, d)$ is the "term frequency", or the number of occurrences of word t in a document d.

If there are only two classes, *pos* and *neg*, maximizing $P(c|d)$ is equivalent to taking the sign of $\ln P(pos|d)/P(neg|d)$, which is a linear combination of the term frequencies $TF(w, d)$. The training consists simply of estimating the probabilities $P(t|c)$ and $P(c)$ from the training documents.

4.3 Data

4.3.1 Document corpus

For experimentation we used a subset of the Reuters-2000 collection [7] and a subset of the Reuters categories. The entire Reuters-2000 collection includes a total of 806,791 documents. For training we used a subset of 118,924 documents comprising about 1600 random positive examples for each Reuters category. For smaller categories we use all the available positive examples [10]. We also restrict the experiments to 16 out of 103 Reuters categories. These are the same categories that were suggested in [10] and selected based on a preliminary document classification experiment that involved a smaller training set of approximately 200 positive examples per category, the complete set of Reuters categories, and a test set of 9,596 documents. The selection took into account the distribution of positive examples in the whole corpus and the precision recall break-even point achieved in the preliminary experiment. Thus, the selected 16 categories (c13, c15, c183, c313, e121, e13, e132, e142, e21, ghea, gobit, godd, gpol, gspo, m14, and m143) are diverse in terms of size and the difficulty they pose as classification problems.

Table 1. Relationship between vector sparsity or density and the total number of retained features. Shows that after retaining 10,000 features from the Odds Ratio ranking we arrive at only 25.7 non-zero components per document vector

Num of Features	Vector Density for the Feature Weighting Method				
	DF	IG	SVM	OR	IDF
1	0.4	0.1	0.0	0.0	0.0
10	2.6	0.5	0.3	0.0	0.0
100	13.0	4.1	2.3	0.0	0.0
1000	43.8	21.3	15.3	0.6	0.0
10000	77.9	51.6	56.1	25.7	0.3
80000	88.3	88.3	88.3	88.3	88.3

Table 2. Number of features in the set, for various levels of vector sparsity or density. Shows that we need to select top 2,593 features from the Odds Ratio ranking to ensure that on document vectors contain 1 non-zero component on average.

Features Per Vector	Feature Set Size for the Feature Weighting Methods				
	DF	IG	SVM	OR	IDF
1	1	22	45	2,593	24,145
10	65	369	591	8,174	65,407
20	203	3,934	1,544	10,570	71,413
30	425	5,988	3,019	12,403	73,517
50	1,430	18,433	8,074	16,768	75,118
80	12,756	60,234	28,637	57,196	75,789

We represent documents using the bag-of-words approach, applying a stop-word filter (from a standard set of 523 stop-words) and ignoring the case of the word surface form. Features that occur less than 4 times in *Train-1600* are removed.

For experiments with the linear SVM classifier we used the SVMLight program (version 3.5) by Thorsten Joachims [4].

5. FRAMEWORK APPLICATION

In this section we show the analysis of the Naïve Bayes performance within the framework described in Section 3. Since some degree of randomness is involved in the construction of synthetic feature rankings, all the results concerning them are averages computed across five such rankings (constructed based on the same characteristic curves). All the results are also macroaveraged across the 16 categories used in our experiments.

5.1 Analysis

5.1.1 Discussion of Sparsity Curves

We observe that there are two natural reference curves: (1) the ranking based on document frequency DF, which sorts the features from the most common to the least common and produces the fastest growing sparsity curve, and (2) the ranking based on IDF, which sorts the features from the least common to most common and has the slowest growing sparsity curve. The other sparsity curves lie between these two extremes. A purely random ranking of features results in a sparsity curve with a constant slope (See Figure 2 and 3).

We may group the resulting sparsity curves into two broad

groups: the quickly growing ones and the slowly growing ones. Information gain (IG) has a quickly growing sparsity curve because the features with high IG should occur in most documents of one class and few documents of the other class. Consequently, very rare features do not have high IG, except if one of the classes is very small. The IG sparsity curve also grows quickly at the end of the ranking, which is where many common words with little relationship to either class are located.

On the other hand, Odds Ratio (OR) has a slowly growing sparsity curve, because many infrequent features have high OR, as long as they do not occur in any negative documents. At the same time, many frequent features occur mostly in negative documents since the negative class is typically larger than the positive and have very low Odds Ratio, resulting in the steep growth of the sparsity curve towards the end of the ranking.

For OR and IG, we are able to understand the sparsity curve from the formulas defining these two rankings. However, for the SVM-based ranking we have no such formula, and consequently the concept of the sparsity curve is particularly interesting because it gives us a better understanding of the SVM feature ranking. The SVM sparsity curve initially grows almost as quickly as that of IG. While IG places most of the infrequent features in the middle of the ranking, SVM tends to give them the lowest scores, relegating them to the end of its ranking.

5.1.2 Cumulative Information Gain Curves

Naturally, the feature ranking based on information gain weights produces the fastest-growing cumulative IG curve (see Figure 5). It is interesting to observe that the feature ranking based on SVM weights also has a fairly quickly growing cumulative IG curve. Thus, it appears that the SVM-based ranking and the IG ranking are similar when characterized by the sparsity curve and the cumulative IG curve. Finding some other feature set characteristics which will distinguish these two rankings better is therefore an interesting problem for further work.

The Odds Ratio ranking has a slower growing curve because it ranks many infrequent features highly while such feature are less likely to have a high information gain. The ranking based on DF has a cumulative IG curve very close to that of the SVM-based ranking, while the ranking based on IDF has an extremely slowly growing IG curve. We do not plot them on the graph in Figure 5 to preserve the readability of the graph.

5.1.3 Synthesized Curves

If we construct a synthetic feature ranking based on the characteristic curves (sparsity curve and cumulative IG curve) of some original ranking, the characteristic curves of the synthetic ranking would ideally be almost identical to those of the original ranking.

Figure 4 and 5 show the characteristic curves of the original and the synthetic rankings. As we can see, the shape of the curves achieved by our synthetic rankings corresponds reasonably well to those of the original rankings, but may in nevertheless differ considerably from the original curve.

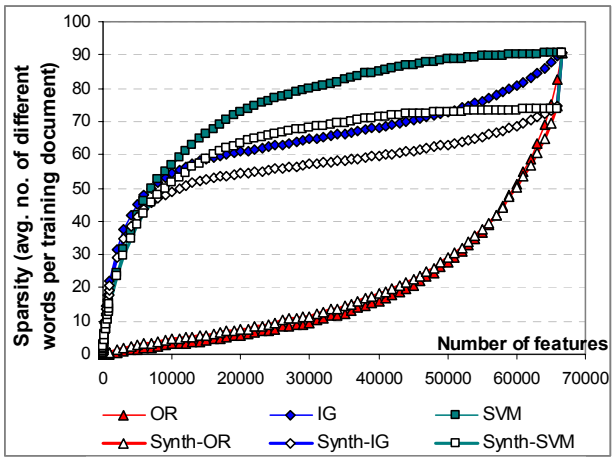

Figure 4. Sparsity curves for Odds Ratio, inf. gain, SVM-based ranking, and the corresponding synthetic rankings.

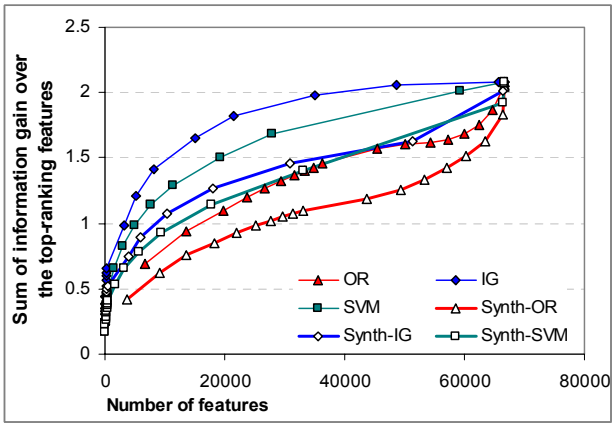

Figure 5. The cumulative information gain curves of feature rankings based on Odds Ratio, information gain, SVM weights, and of the corresponding synthetic feature rankings.

This is a deficiency of our version of the synthetic ranking construction scheme presented in Sec. 3.2, which will be addressed in our future work. If we apply one constraint only, e.g. the sparsity curve, it is not difficult to obtain a synthetic ranking whose curve matches the original one almost perfectly. However, the resulting rankings perform poorly on the classification task (see Section 5.3).

5.2 Document Classification Performance

In order to evaluate the performance of different feature rankings, we calculate the macroaveraged F_1 measure across the 16 categories. Figure 6 and 7 show the F_1 measure as a function of sparsity for easier readability and comparison than when N is used as independent variable, In N based graphs, rankings such as Odds Ratio, with its preference for rare features, would have a very low performance for a wide range of 'small' values of N.

Of the original rankings, the one based on the SVM normal weights is the most successful while the OR (Odds Ratio) ranking performs slightly better than IG (information gain.

The synthetic rankings and corresponding performance curves are

designated by prefix Synth-. Unsurprisingly, Synth-IG performs very similarly to the original IG since it is completely determined by the one of the constraints, the cumulative IG curve.

On the other hand, Synth-SVM has a performance curve similar to IG (and Synth-IG) but much worse than the original SVM. This similarity is probably due to the fact that the shape of both the sparsity and the cumulative IG curve of the original SVM and IG rankings are similar. At the same time, the gap between the performance of Synth-SVM and the original SVM ranking remains large, which shows that a part of original SVM ranking success stems from properties of the feature set that has not been capture by the two characteristic curves used in our experiments.

As for Synth-OR, although its performance is much below that of the original OR ranking, the shape of its performance curve is remarkably similar to the original OR (Figure 6). This suggests that, although the sparsity and IM constraints do not approximate the OR ranking sufficiently well, the resulting ranking preserves the same increase in performance with the growing feature set.

5.2.1 One constraint vs two constraints

For the sake of comparison, Figure 7 shows the performance of synthetic rankings obtained by taking only the sparsity curves into account, while disregarding the cumulative IG curves. We expect a loss in classification performance as no information about class membership is included and enforced by the sparsity constraint. The results are still informative. They show to what extent the performance of a ranking is influenced by its sparsity curve.

For example, SVM and IG have similarly-shaped sparsity curves (Figure 4) and the corresponding synthetic rankings have very similar performance curves. On the other hand, OR has a similarly-shaped sparsity curve as IDF, and the performance curves of Synth-OR, Synth-IDF and IDF are almost exactly the same. This shows that the shape of a sparsity curve by itself already has an effect on the performance of a feature ranking.

5.3 Comparison of Synthetic and Original Feature Rankings

If we try to synthesize a feature ranking that approximately exhibits the same characteristic curves as another ranking, how similar are the two rankings themselves? To answer this question we compare two feature rankings using the following method. Let $A(S)$ be the set of features from the original ranking, required to attain the sparsity S of our training documents. Analogously, let $B(S)$ be the set of features required to satisfy the sparsity S but for the synthetic ranking. We compute the size of the *relative intersection* of the two sets $|A(S) \cap B(S)|/|A(S) \cup B(S)|$ (common vs all the features) as a measure of the overlap between the feature sets that correspond to the sparsity S. By varying S we obtain the overlap curve; if the two rankings are the same, the curve would have a constant value of 1 for all S.

The chart on in Figure 8 shows the resulting curves for Odds Ratio, IG, and the SVM based ranking. We see considerable discrepancies between the original and the synthetic rankings, resulting in relative intersection sizes much less than 1.

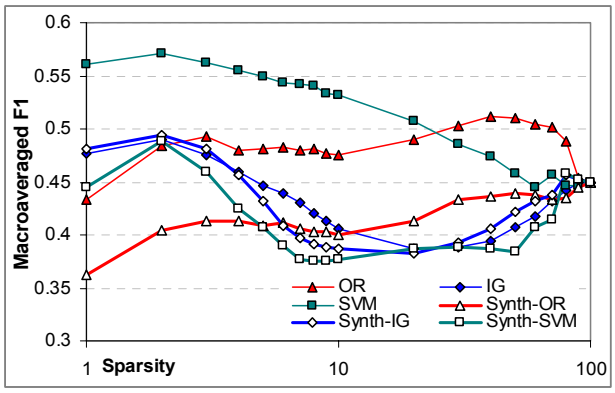

Figure 6. Macroaveraged F_1 measure for the Naïve Bayes classifier using different feature ranking schemes. For clarity, the y-axis shows only the range from 0.3 to 0.6.

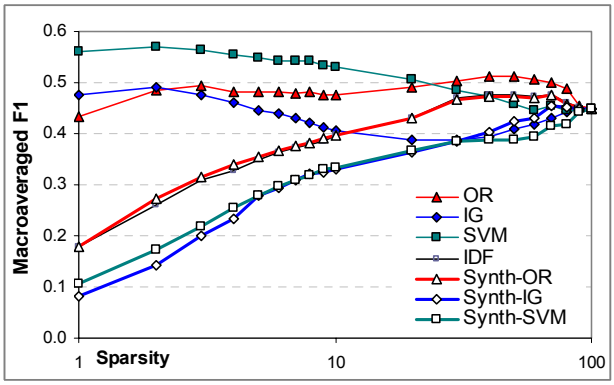

Figure 7. Performance of synthetic rankings based *only* on the sparsity curves, ignoring the cumulative IG curves.

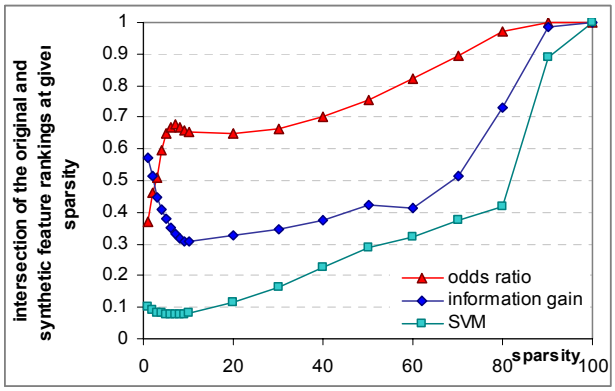

Figure 8. A comparison of the original feature rankings with synthesized ranking

This shows that the feature rankings are only partially characterized by the characteristic curves employed in our experiments, i.e., the sparsity curve and the cumulative information gain curve.

The synthetic counterpart to Odds Ratio has a fairly large intersection with the original Odds Ratio ranking. We note that for the OddsRatio, even at low sparsity levels, there are lots of features involved (See Tables 1 and 2). This may make it more

probably that the relative intersection will be larger. We also note that the intersection between the SVM ranking and its synthetic counterpart is relatively low, indicating that SVM ranking is the most sophisticated and least accurately characterized by the two characteristic curves used in these experiments.

We are aware that some of the set discrepancies are caused by the imperfections in our algorithm for synthesizing feature ranking for a given constraint. This will be addressed in our future work.

6. CONCLUSIONS

In this paper we introduce a framework for analyzing the feature ranking schemes and their impact on the performance of a given classifier. The properties of both the term weighing schemes and the classification performance are essentially 'projected' onto the space of characteristic functions for the given features sets. By synthesizing alternative feature rankings that satisfy the same constraints we can probe to which degree each of the identified properties explain the performance of the classifier or determines the ranking associated with the given feature weighting scheme.

We apply this method to the Naïve Bayes classifier, and five distinct feature weighting schemes. We explore two sets of statistics as a vehicle for characterizing the feature sets: the sparsity and the cumulative information gain curves. We demonstrate that the combined constraints yield feature rankings which exhibits a higher accuracy in approximating the original ranking, associated with the feature weighting scheme. On the other side they also improve the classification performance.

Our approach is unique in its attempt to provide a framework for increasing the understanding of the interaction between the feature selection and classification methods. With a suitable set of characterizing functions for feature rankings it might be possible to synthesize feature rankings that resemble and perform equally well as feature weighting based on sophisticated methods like SVM.

As our experiments show, the sparsity curves and the cumulative IG curves already provide an interesting characterization of feature rankings, but are not sufficient to synthesize good approximations of the existing rankings. Thus one of the main directions for further work is to investigate the set of characteristic functions that could serve this purpose and their mutual relation ship. The underlying assumption is that feature scoring and classification functions over the ranked feature sets are well behaved and thus decomposable into elementary functions that characterize the features sets.

Another objective of the further work is to improve the algorithm for constructing synthetic feature rankings to satisfy a given set of constraints. The improved algorithm should be able to obtain rankings that follow the constraints more tightly than we can achieve now. Ideally, one would be able to simply synthesize a good feature ranking based on the approximate shape of a few characteristic curves. However, for this approach to be useful in practice, properties of good rankings should have roughly similar characteristic curves across datasets and constraints. Thus, an important topic for further work is to investigate to what extent the shapes of such characteristic curves is stable across datasets and categories.

7. REFERENCES

[1] S. Dumais, J. Platt, D. Heckerman, M. Sahami: *Inductive learning algorithms and representations for text categoryzation*. Proc. 1998 CIKM Conf (Bethesda, Maryland, USA, Nov 3–7, 1998), pp. 148–155.

[2] T. Joachims: *Text categorization with support vector machines: learning with many relevant features*. Proc. 10th European Conf. on Machine Learning (Chemnitz, Germany, April 21–23, 1998). LNCS vol. 1398, pp. 137–142.

[3] D. Mladenić: *Feature subset selection in text-learning*. Proc. 10th European Conf. on Machine Learning (Chemnitz, Germany, April 21–23, 1998). LNCS vol. 1398, pp. 95–100.

[4] T. Joachims: *Making large-scale support vector machine learning practical*. In: B. Schölkopf, C. J. C. Burges, A. J. Smola (Eds.): *Advances in kernel methods: Support vector learning*, The MIT Press, 1999, pp. 169–184

[5] D. Mladenić, M. Grobelnik: *Feature selection for unbalanced class distribution and Naive Bayes*. Proc. of the 16th Int. Conf. on Machine Learning (Bled, Slovenia, June 27–30, 1999), pp. 258–267.

[6] Y. Yang, J. O. Pedersen: *A comparative study on feature selection in text categorization*. Proc. of the 14th Int. Conf. on Machine Learning (Nashwille, Tennessee, USA, July 8–12, 1997), pp. 412–420.

[7] Reuters Corpus, Volume 1, English Language, 1996-08-20 to 1997-08-19. Released in November 2000. Available at http://about.reuters.com/researchandstandards/corpus/.

[8] Cortes, C., Vapnik, V.: *Support-vector networks*. Machine Learning, 20(3):273–297, September 1995.

[9] Ng, H.T., Goh, W.B., Low, K.L. Feature selection, perceptron learning, and a usability case study for text categorization. In Proc. 20th ACM SIGIR Conf. (Philadelphia, Pennsylvania, USA, July 27–31, 1997), ACM Press, pp. 67-73.

[10] Brank, J., Grobelnik, M., Milić-Frayling, N., Mladenić, D. Feature selection using support vector machines. In Proc. 3rd Int. Conf. on Data Mining Methods and Databases for Engineering, Finance, and Other Fields (Bologna, Italy, September 2002).

[11] Sindhwani, V., Bhattacharya, P., Rakshit, S. Information theoretic feature crediting in multiclass Support Vector Machines. In Proceedings of the 1st SIAM Int. Conf. on Data Mining. SIAM, Philadelphia, 2001.

[12] McCallum, A., Nigam, K. A comparison of event models for naive Bayes text classification. In: Learning from Text Categorization: Papers from the AAAI Workshop (Madison, Wisconsin, 1998), TR WS-98-05, AAAI Press, pp. 41–48.

[13] Mladenić, D., Brank, J., Grobelnik, M., Milić-Frayling, N. Feature selection using linear classifier weights: Interaction with classification models. Proc. 27th ACM SIGIR Conference (Sheffield, UK, July 25–29, 2004), pp. 243–241.