Journal of Machine Learning Research 3 (2003) 1289-1305 Submitted 5/02; Published 3/03

An Extensive Empirical Study of Feature Selection Metrics for Text Classification

George Forman GFORMAN@HPL.HP.COM
Hewlett-Packard Labs
Palo Alto, CA, USA 94304

Editors: Isabelle Guyon and André Elisseeff

Abstract

Machine learning for text classification is the cornerstone of document categorization, news filtering, document routing, and personalization. In text domains, effective feature selection is essential to make the learning task efficient and more accurate. This paper presents an empirical comparison of twelve feature selection methods (e.g. Information Gain) evaluated on a benchmark of 229 text classification problem instances that were gathered from Reuters, TREC, OHSUMED, etc. The results are analyzed from multiple goal perspectives—accuracy, F-measure, precision, and recall—since each is appropriate in different situations.

The results reveal that a new feature selection metric we call 'Bi-Normal Separation' (BNS), outperformed the others by a substantial margin in most situations. This margin widened in tasks with high class skew, which is rampant in text classification problems and is particularly challenging for induction algorithms.

A new evaluation methodology is offered that focuses on the needs of the data mining practitioner faced with a single dataset who seeks to choose one (or a pair of) metrics that are most *likely* to yield the best performance. From this perspective, BNS was the top single choice for all goals except precision, for which Information Gain yielded the best result most often. This analysis also revealed, for example, that Information Gain and Chi-Squared have correlated failures, and so they work poorly together. When choosing optimal pairs of metrics for each of the four performance goals, BNS is consistently a member of the pair—e.g., for greatest recall, the pair BNS + F1-measure yielded the best performance on the greatest number of tasks by a considerable margin.

Keywords: support vector machines, document categorization, ROC, supervised learning

1 Introduction

The potential is great for machine learning to categorize, route, filter and search for relevant information. For example, to build and populate a Web portal or news directory, a person would identify a modest number of training examples for each category, and then an induction algorithm can learn the pattern and identify additional matches to populate the directory. However, as problem sizes continue to scale up with the explosive growth of the Internet, essential research is required to further improve classification efficiency and accuracy. One leg of this research is in induction algorithms, where Support Vector Machines (SVM) have recently shown great promise (e.g., Yang & Liu, 1999). The other leg, our topic of study here, is in feature selection—no degree of clever induction can make up for a lack of predictive signal in the input features.

In text classification, one typically uses a 'bag of words' model: each position in the input feature vector corresponds to a given word or phrase. For example, the occurrence of the word 'free' may be a useful feature in discriminating spam email. The number of potential words often exceeds the number of training documents by more than an order of magnitude. Feature selection is necessary to make large problems computationally efficient—conserving computation, storage

© 2003 Hewlett-Packard

and network resources for the training phase and for every future use of the classifier. Further, well-chosen features can improve classification accuracy substantially, or equivalently, reduce the amount of training data needed to obtain a desired level of performance. Opinions differ as to why this is so, but it is frequently acknowledged.

This paper presents an empirical study of twelve feature selection metrics evaluated on a benchmark of 229 text classification problem instances that originated from Reuters, OHSUMED, TREC, etc. (Han & Karypis, 2000). Our primary focus is on obtaining the best overall classification performance regardless of the number of features needed to obtain that performance. We also analyze which metrics excel when only a very small number of features is selected, which is important for situations where machine resources are severely limited, fast classification is needed, or large scalability is demanded. We analyze the results from each of the perspectives of accuracy, precision, recall, and F-measure, since each serves different purposes.

In text classification problems, there is typically a substantial class distribution skew, and it worsens as the problem size scales up. For example, in selecting news articles that best match one's personalization profile, the positive class of interest contains many fewer articles than the negative background class, esp. if the background class contains all news articles posted on the Internet worldwide. For multi-class problems, the skew increases with the number of classes. The skew of the classification tasks used in this study is 1:31 on average, and 4% exceed 1:100. While much machine learning research does not consider such extreme skews, the future holds classification tasks with ever increasing skews. For example, in an information retrieval setting, the user may possess only a single positive example to contrast against a large database of presumably negative training examples.

High class skew presents a particular challenge to induction algorithms, which are hard pressed to beat the high accuracy achieved by simply classifying everything as the negative majority class. We hypothesize that feature selection should then be relatively more important in difficult, high-skew situations. This study also contrasts the performance under high-skew and low-skew situations, validating this hypothesis.

Finally, we introduce a novel analysis that is focused on a subtly different goal: to give guidance to the data mining practitioner about which feature selection metric or combination is *most likely* to obtain the best performance for the *single given* dataset at hand, supposing their text classification problem is drawn from a distribution of problems similar to that studied here.

The results on these benchmark datasets showed that the well-known Information Gain metric is a decent choice if one's goal is precision, but for accuracy, F-measure, and in particular for recall, a new feature selection metric we call 'Bi-Normal Separation,' showed outstanding performance. In high-skew situations, its performance is exceptional.

Scope: In this study, we consider each binary class decision as a distinct problem instance and select features for it separately. This is the natural setting for 2-class problems, e.g. in identifying spam vs. valuable email. This is also an important subcomponent for good multi-class feature selection for two different types of problems: (1) '1-of-m' multi-class problems, e.g. determining where to file a new item for sale in the large Ebay.com classified ad categories (2) 'n-of-m' problems (aka *topic* or *keyword identification*) where a single set of features for all m problems is used. In such situations, selecting features separately for each class (vs. altogether) can extend the reach of induction algorithms to greater problem sizes with greater levels of class skew.

The choice of the induction algorithm is not the object of study here. Previous studies have shown SVM (Schölkopf & Smola, 2002) to be a consistent top performer (e.g., Yang & Liu, 1999; Joachims, 1998; Dumais *et al.,* 1998), and a pilot study comparing the use of the popular Naïve Bayes algorithm, logistic regression, and C4.5 decision trees confirmed its superiority. We use the default parameters and a linear kernel; later studies may wish to explore the interaction of feature selection and model tuning.

1.1 Related Work

For context, we mention that a large number of studies on feature selection have focused on non-text domains. These studies typically deal with much lower dimensionality, and often find that wrapper methods perform best (e.g., Kohavi & John, 1997). Wrapper methods, such as sequential forward selection or genetic search, perform a search over the space of all possible subsets of features, repeatedly calling the induction algorithm as a subroutine to evaluate various subsets of features. For large scale problems, however, wrapper methods are often impractical, and instead feature scoring metrics (filter methods) are used independently on each feature. This paper is only concerned with feature scoring metrics; nevertheless, we note that advances in scoring methods should be welcome to wrapper techniques for use as heuristics to guide their search more effectively.

Previous feature selection studies for *text* domain problems have been a great help in providing guidance and motivation for this study, which features a more extensive variety of metrics, a larger set of benchmark problems, and one of the best induction algorithms of late, support vector machines. For example, the valuable study by Yang and Pedersen (1997) considered five feature selection metrics on the standard Reuters dataset and OHSUMED. It did not consider SVM, which they later found to be superior to the algorithms they had studied, LLSF and kNN (Yang and Liu, 1999). The question remains then: do their findings generalize to SVM? The study by Mladenic and Grobelnik (1999) promoted Odds Ratio over a wide variety of metrics, but for a different dataset and only for the Multinomial Naïve Bayes algorithm. Various other studies are incomparable for various reasons, such as considering disjoint goals—accuracy vs. F-measure, and best performance vs. best performance at a small number of features. The need remains to compare these disjoint sets of metrics on a common and substantial dataset with the state-of-the-art induction algorithm and to analyze the results from each prospective goal. This study recommends feature selection strategies for varied situations, e.g. different tradeoffs between precision and recall, for when resources are tight, and for highly skewed datasets.

Furthermore, most text feature selection studies consider the problem of selecting one set of features for 1-of-m or n-of-m multi-class problems. This fails to explore the best possible accuracy obtainable for any single class, which is especially important for high class skew, for which we present an explicit analysis.

2 Feature Selection Filtering Methods

The overall feature selection procedure is to score each potential feature according to a particular feature selection metric, and then take the best k features. Scoring involves counting the occurrences of a feature in training positive- and negative-class training examples separately, and then computing a function of these.

Before we enumerate the feature selection metrics we studied, we briefly describe filters that are commonly applied prior to using the feature selection metric, which can have a substantial bearing on the final outcome.

First, rare words may be eliminated, on the grounds that they are unlikely to be present to aid in future classifications. For example, words occurring two or fewer times may be removed. Word frequencies typically follow a Zipf distribution: the frequency of each word's occurrence is proportional to $1/\text{rank}^p$, where *rank* is its rank among words sorted by frequency, and p is a fitting factor close to 1.0 (Miller 1958). Easily half of the total number of distinct words may occur only a single time, so eliminating words under a given low rate of occurrence yields great savings. The particular choice of threshold value can have an effect on accuracy, which we demonstrate in the discussion section. If we eliminate rare words based on a count from the whole dataset *before* we split off a training set, we have leaked some information about the test set to the training phase. Without expending a great deal more resources for cross-validation studies, this research practice is unavoidable, and is acceptable in that it does not use the class labels of the test set.

Additionally, overly common words, such as 'a' and 'of', may also be removed on the grounds that they occur so frequently as to not be discriminating for any particular class. Common words can be identified either by a threshold on the number of documents the word occurs in, e.g. if it occurs in over half of all documents, or by supplying a *stopword* list. Stopwords are language-specific and often domain-specific. Depending on the classification task, they may run the risk of removing words that are essential predictors, e.g. the word 'can' is discriminating between 'aluminum' and 'glass' recycling.

It is also to be mentioned that the common practice of *stemming* or *lemmatizing*—merging various word forms such as plurals and verb conjugations into one distinct term—also reduces the number of features to be considered. It is properly, however, a *feature engineering* option.

An ancillary feature engineering choice is the representation of the feature value. Often a Boolean indicator of whether the word occurred in the document is sufficient. Other possibilities include the count of the number of times the word occurred in the document, the frequency of its occurrence normalized by the length of the document, the count normalized by the inverse document frequency of the word. In situations where the document length varies widely, it may be important to normalize the counts. For the datasets included in this study, most documents are short, and so normalization is not called for. Further, in short documents words are unlikely to repeat, making Boolean word indicators nearly as informative as counts. This yields a great savings in training resources and in the search space of the induction algorithm. It may otherwise try to discretize each feature optimally, searching over the number of bins and each bin's threshold. For this study, we selected Boolean indicators for each feature. This choice also widens the choice of feature selection metrics that may be considered, e.g. Odds Ratio deals with Boolean features, and was reported by Mladenic and Grobelnik (1999) to perform well.

A final choice in the feature selection policy is whether to rule out all negatively correlated features. Some speculate that classifiers built from positive features only may be more robust in the special situation where the background class may shift and retraining is not an option, although this has yet to be validated. Additionally, some classifiers work primarily with positive features, e.g. the Multinomial Naïve Bayes model, which has been shown to be both better than the traditional Naïve Bayes model (McCallum & Nigam, 1998), and considerably inferior to other induction methods for text classification (e.g., Yang & Liu, 1999; Dumais *et al.*, 1998). Negative features are numerous, given the large class skew, and quite valuable in practical experience: For example, when scanning a list of Web search results for the author's home page, a great number of hits on George Foreman the boxer show up and can be ruled out strongly via the words 'boxer' and 'champion,' of which the author is neither. The importance of negative features is empirically confirmed in the evaluation.

2.1 Metrics Considered

Here we enumerate the feature selection metrics we evaluated. Their formulae are shown in Table 1, including footnotes about their properties. In the interest of brevity, we omit their varied mathematical justifications that have appeared in the literature (e.g., Mladenic & Grobelnik, 1999; Yang & Pedersen, 1997). In the following subsection, we show a novel graphical analysis that reveals the widely different decision curves they induce. Paired with an actual sample of words, this yields intuition about their empirical behavior.

Commonly Known Metrics:

Chi: Chi-Squared is the common statistical test that measures divergence from the distribution expected if one assumes the feature occurrence is actually independent of the class value. As a statistical test, it is known to behave erratically for very small expected counts, which are common in text classification both because of having rarely occurring word features, and sometimes because of having few positive training examples for a concept.

Name	Description	Formula
Acc	Accuracy	$tp - fp$
Acc2	Accuracy balanced[†]	$\mid tpr - fpr \mid$
BNS	Bi-Normal Separation[†]	$\mid F^{-1}(tpr) - F^{-1}(fpr) \mid$ where F is the Normal c.d.f.
Chi	Chi-Squared[‡]	$t(tp, (tp+fp)P_{pos}) + t(fn, (fn+tn)P_{pos}) + t(fp, (tp+fp)P_{neg}) + t(tn, (fn+tn)P_{neg})$ where $t(count, expect) = (count - expect)^2 / expect$
DFreq	Document Frequency[†‡o]	$tp + fp$
F1	F_1-Measure	$\frac{2\,recall\ precision}{(recall + precision)} = \frac{2tp}{(pos + tp + fp)}$
IG	Information Gain[†‡]	$e(pos, neg) - [\, P_{word}\, e(tp, fp) + P_{\overline{word}}\, e(fn, tn)\,]$ where $e(x, y) = -\frac{x}{x+y}\log_2\frac{x}{x+y} - \frac{y}{x+y}\log_2\frac{y}{x+y}$
OddN	Odds Ratio Numerator	$tpr\,(1 - fpr)$
Odds	Odds Ratio[†]	$\frac{tpr\,(1-fpr)}{(1-tpr)\,fpr} = \frac{tp\ tn}{fp\ fn}$
Pow	Power	$(1 - fpr)^k - (1 - tpr)^k$ where k=5
PR	Probability Ratio	tpr / fpr
Rand	Random[‡o]	random()

[†] Acc2, BNS, DFreq, IG, and Odds select a substantial number of negative features.
[‡] Chi, IG, DFreq, and Rand also generalize for multi-class problems.
[o] DFreq and Rand do not require the class labels.

Notation:

tp: true positives = number of positive cases containing word
fp: false positives = number of negative cases containing word
pos: number of positive cases = *tp* + *fn*
neg: number of negative cases = *fp* + *tn*
tpr: sample true positive rate = *tp* / *pos*
fpr: sample false positive rate = *fp* / *neg*
precision = *tp* / (*tp*+*fp*)
fn: false negatives
tn: true negatives
P_{pos} = *pos* / *all*
P_{neg} = *neg* / *all*
P_{word} = (*tp*+*fp*) / all
$P_{\overline{word}}$ = 1-P(word)
recall = *tpr*

Note: Metrics such as BNS, Chi and IG are naturally symmetric with respect to negatively correlated features. For the metrics that devalue all negative features, we invert any negative feature, i.e. *tpr'* = 1 – *tpr* and *fpr'* = 1 – *fpr*, without reversing the classes. Hence, without loss of generality, *tpr* > *fpr*.

Table 1. Feature Selection Metrics

IG: Information Gain measures the decrease in entropy when the feature is given vs. absent. Yang and Pederson (1997) reported IG and Chi performed best in their multi-class benchmarks. In contrast, they found Mutual Information and Term Strength performed terribly, and so we do not consider them further. IG has a generalized form for nominal valued attributes.

Odds: Odds Ratio reflects the odds of the word occurring in the positive class normalized by that of the negative class. It has been used for relevance ranking in information retrieval. In the study by Mladenic and Grobelnik (1999), it yielded the best F-measure for Multinomial Naïve Bayes, which works primarily from positive features. To avoid division by zero, we add one to any zero count in the denominator.

PR: (Log) Probability Ratio is the sample estimate probability of the word given the positive class divided by the sample estimate probability of the word given the negative class. It induces the same decision surface as the log probability ratio, $\log(tpr/fpr)$ (Mladenic and Grobelnik, 1999), and is faster to compute. Since it is not defined at $fpr=0$, we explicitly establish a preference for features with higher tp counts along the axis by substituting $fpr'=1e\text{-}8$.

DFreq: Document Frequency simply measures in how many documents the word appears. Since it can be computed without class labels, it may be computed over the entire test set as well. Selecting frequent words will improve the chances that the features will be present in future test cases. It performed much better than Mutual Information in the study by Yang and Pedersen, but was consistently dominated by IG and Chi (which, they point out, each have a significant correlation with frequent terms).

Additional Metrics:

Rand: Random ranks all features randomly and is used as a baseline for comparison. Interestingly, it scored highest for precision in the study by Mladenic and Grobelnik (1999), although this was not considered valuable because its recall was near zero.

F1: F_1-measure is the harmonic mean of the precision and recall. This metric is motivated because in many studies the F-measure is the ultimate measure of performance of the classifier. Note that it focuses on the positive class, and hence negative features, even if inverted, are devalued compared to positive features. This is ultimately its downfall as a feature selection metric, esp. for precision.

OddN: Odds Ratio Numerator is the numerator of Odds Ratio.

Acc: Accuracy estimates the expected accuracy of a simple classifier built from the single feature, i.e. P(1 for + class and 0 for – class) = $P(1|+)\ P_{pos} + P(0|-)\ P_{neg} = tpr\ P_{pos} + (1\text{-}fpr)\ P_{neg}$, which simplifies to the simple decision surface $tp - fp$. Note that it takes the class skew into account. Since P_{neg} is large, fpr has a strong influence.

Acc2: Accuracy2 is similar, but supposes the two classes were balanced in the equation above, yielding the decision surface equivalent to $tpr - fpr$. This removes the strong preference for low fpr. It induces the same decision surface as the 'ExpProbDiff(W)' metric studied by Mladenic and Grobelnik (1999).

BNS: Bi-Normal Separation is a new feature selection metric we defined as $F^{-1}(tpr) - F^{-1}(fpr)$, where F^{-1} is the standard Normal distribution's inverse cumulative probability function (a.k.a. *z-score*). To avoid the undefined value $F^{-1}(0)$, zero is substituted by 0.0005, half a count out of 1000.

For intuition, suppose the occurrence of a given feature in each document is modeled by the event of a random Normal variable exceeding a hypothetical threshold. The prevalence rate of the feature corresponds to the area under the curve past the threshold. If the feature is more prevalent in the positive class, then its threshold is further from the tail of the distribution than that of the negative class. (Refer to Figure 1.) The BNS metric measures the separation between these two thresholds.

Figure 1. Two views of Bi-Normal Separation using the Normal probability distribution: (left) Separation of thresholds. (right) Separation of curves (ROC analysis).

An alternate view is motivated by ROC threshold analysis: The metric measures the horizontal separation between two standard Normal curves where their relative position is uniquely prescribed by *tpr* and *fpr*, the area under the tail of each curve (cf. a traditional hypothesis test where *tpr* and *fpr* estimate the center of each curve). The BNS distance metric is therefore proportional to the area under the ROC curve generated by the two overlapping Normal curves, which is a robust method that has been used in the medical testing field for fitting ROC curves to data in order to determine the efficacy of a treatment. Its justifications in the medical literature are many and diverse, both theoretical and empirical (Hanley, 1988; Simpson & Fitter, 1973), such as fitting well both actual and artificial data that violates the Normal assumption.

Pow: Power, although theoretically unmotivated, is considered because it prefers frequent terms (Yang & Pedersen, 1997), aggressively avoids common *fp* words, and can generate a variety of decision surfaces given parameter *k*, with higher values corresponding to a stronger preference for positive words. We chose *k*=5 after a pilot study.

2.2 Graphical Analysis

In order to gain a more intuitive grasp of the feature selection biases of these various metrics, we present graphs of example decision boundaries these metrics induce. We illustrate with a prototypical binary text classification problem from the Cora dataset (see 0): 50 research papers on probabilistic machine learning methods vs. 1750 other computer science papers. Refer to the ROC graph in Figure 2. The dots represent the true positive and false positive counts for each potential word feature, gathered from the titles and abstracts of the papers. The distribution of the words looks similar for many text classification tasks.

There are no strongly predictive features in the top left or bottom right corners, and due to the high class skew, there are many more negatively correlated words along the x-axis than positive words along the y-axis. Very few words have high frequency and they tend to be non-predictive, i.e. they stay close to the diagonal as they approach the upper right corner. This partly supports the practice of eliminating the most frequent words—the vertical dotted line near 400 depicts a cut-off threshold that eliminates words present in >¼ of all documents—but note that this eliminates merely 28 words out of 12,500. Likewise, the practice of eliminating stopwords is of limited usefulness. The diamonds mark 97 words that appear on a stopword list of 152 English words. These do tend to be non-predictive, but eliminating them removes only a tiny fraction of the many potential words, and most feature selection metrics would discard them anyway.

For each of the feature selection metrics depicted on the graph, we scored all word features and determined the score threshold that selects exactly 100 words. We then plotted the isocline having that threshold value. For example, the BNS threshold is 1.417, selecting high-scoring features above its upper curve and below its symmetric lower curve. We see that the isoclines for Odds Ratio and BNS go all the way to the origin and top right corner, while IG and Chi progressively cut off the top right—and symmetrically the bottom left—eliminating many negative features that Odds and BNS include. In contrast, BNS selects fewer positive words than IG, for example. PR is the most extreme, selecting only positive features, since there are no words in the bottom right. The DFreq line selects 100 words to its right (after we removed the 'too common' words to the right of the dotted line); observe that this swath selects mostly non-predictive and negative features. Refer to Appendix C for graphs of the other metrics.

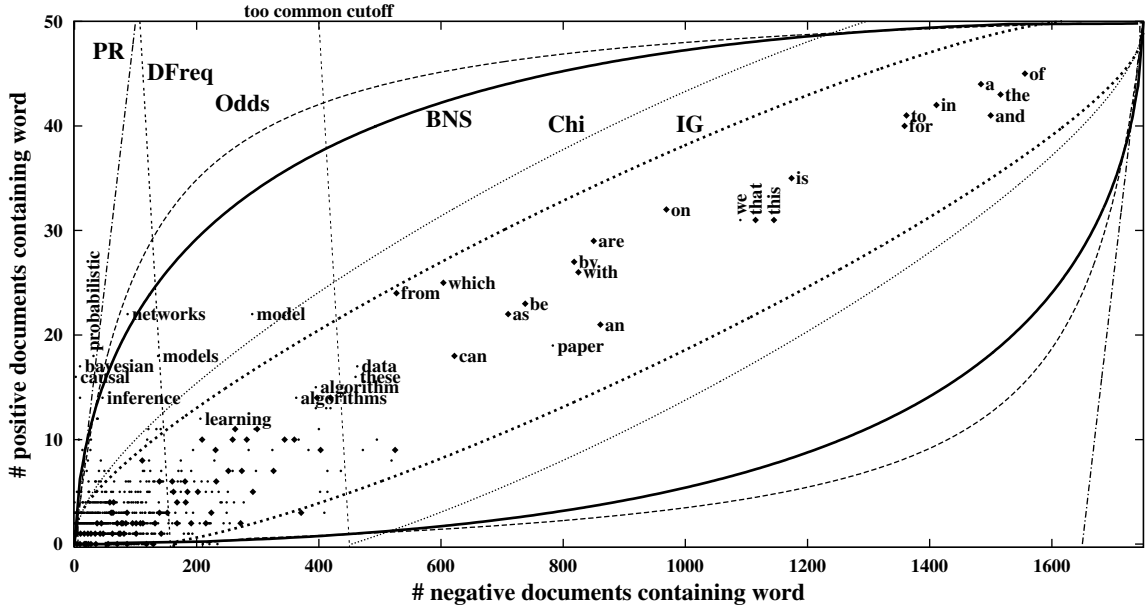


Figure 2. Decision boundary curves for the feature selection metrics Probability Ratio, Document Frequency, Odds Ratio, Bi-Normal Separation, Chi-Squared, and Information Gain. Each curve selects the 'best' 100 words, each according to its view, for discriminating abstracts of probabilistic reasoning papers from others. Dots represent actual words, and many of the 12,500 words overlap near the origin.

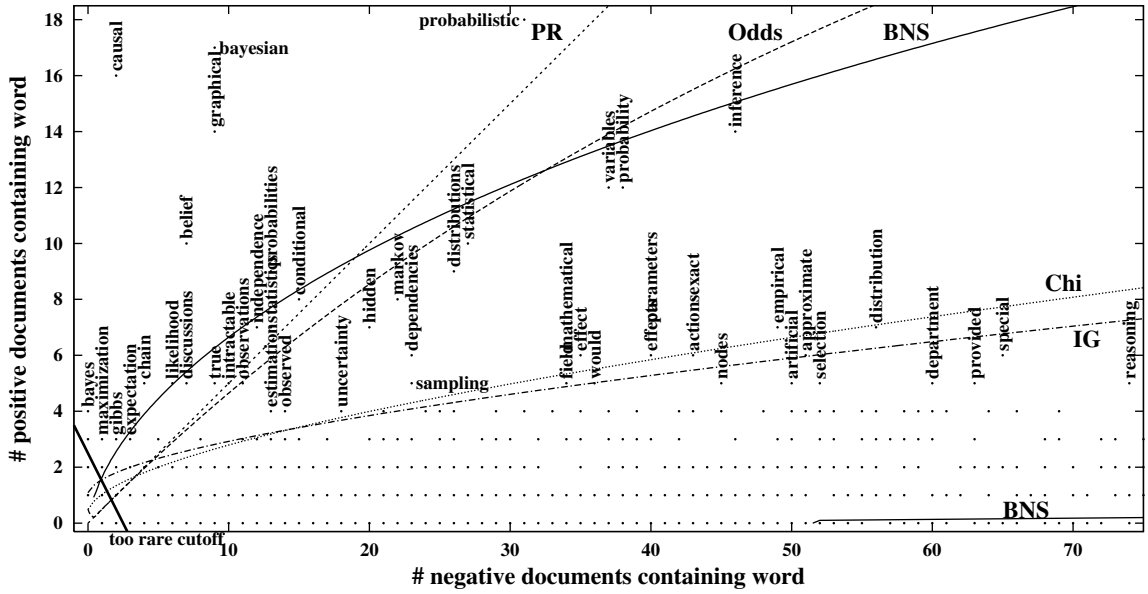


Figure 3. Zoomed-in version of Figure 2, detailing where most words occur.

Since word frequencies tend toward a Zipf distribution, most of the potential word features appear near the origin. This implies that feature selection is most sensitive to the shape of the decision boundary in these dense regions. Figure 3 shows a zoomed-in view. Here there are many words with identical *tp* and *fp* counts, which overlay a single dot in the visualization. For example, the positive words 'bayes,' 'causation' and 'posterior' collide at (0,3), and there are 1536 distinct words at point (2,0). The bold diagonal line near the origin shows a rare word cutoff

of < 3 occurrences, which eliminates 7333 words for this dataset. This represents substantial resource savings (~60%) and the elimination of fairly uncertain words that are unlikely to re-occur at a rate that would be useful for classification.

3 Experimental Method

Performance measures: A number of studies on feature selection, primarily those outside the text domain, have focused on accuracy. Accuracy and error rate can be weak indicators of performance when there is substantial class skew. For example, some classes in the TREC datasets are represented by seven positive examples out of 927, giving the trivial majority classifier an accuracy of 99.2%, and thereby compressing the range of interesting values to the remaining 0.8%. Therefore, the information retrieval community, which is often faced with much greater class skews than the machine learning community has traditionally addressed, prefers the measures *precision* (the percentage of items classified as positive that actually are positive), *recall* (the percentage of positives that are classified as positive), and the *F-measure* (their harmonic average, see Table 1). While several studies have sought solely to maximize the F-measure, there are common situations where precision is to be strongly preferred over recall, e.g. when the cost of false positives is high, such as mis-filtering a legitimate email as spam. Precision should also be the focus when delivering Web search results, where the user is likely to look at only the first page or two of results; the retrieval strategy might be switched dynamically if the user ends up exhausting the result list. Finally, there are situations where accuracy is the most appropriate measure, even when there is high class skew, e.g. equal misclassification costs. For these reasons, we analyze performance for each of the four performance goals.

There are two methods for averaging the F-measure over a collection of 2-class classification problems. One is the *macro-averaged F-measure*, which is the traditional arithmetic mean of the F-measure computed for each problem. Another is the *micro-averaged F-measure*, which is an average weighted by the class distribution. The former gives equal weight to each problem, and the latter gives equal weight to each document classification (which is equivalent to overall accuracy for a 1-of-m problem). Since highly skewed, small classes tend to be more difficult, the macro-averaged F-measure tends to be lower. (The two are equal for the uniform multi-class distribution.) We focus on macro-averaging because we are interested in average performance across different problems, without regard to the training set size of each. To measure performance for a given problem instance, we use 4-fold stratified cross-validation, and take the average of 5 such runs. 0 contains pseudo-code to describe the complete measurement procedure.

A data mining practitioner has a different goal in mind: to choose a feature selection technique that maximizes their *chances* of having the best metric *for their single dataset of interest.* Supposing the classification problems in this study are representative of problems encountered in practice, we compute for each metric, the percentage of problem instances for which it was optimal, or within a given error tolerance of the best method observed for that instance. Furthermore, we use this framework to determine the best *pair* of techniques to try in combination. This analysis has greater practical value than the practice of wins/ties/losses because of correlated failures, as discussed below.

Induction Algorithm: We performed a brief pilot study using a variety of classifiers, including Naïve Bayes, C4.5, logistic regression and SVM with a linear kernel (each using the WEKA open-source implementation with default parameters). The results confirmed previous findings that SVM is an outstanding method (Yang & Liu, 1999; Joachims, 1998; Dumais et al., 1998), and so the remainder of our presentation uses it alone. It is an interesting target for feature selection because no comparative text feature selection studies have yet considered it, and its use of features is entirely along the decision boundary between the positive and negative classes, unlike many traditional induction methods that model the density. We note that the traditional

Naïve Bayes model fared better than C4.5 for these text problems, and that it was fairly sensitive to feature selection, having its performance peak at a much lower number of features selected.

Datasets: We were fortunate to obtain a large number of text classification problems in preprocessed form made available by Han and Karypis (2000), the details of which are laid out in their paper and in 0. We added a dataset of computer science paper abstracts gathered from Cora.whizbang.com that were categorized into 36 classes, each containing 50 training examples to control the class skew. Taken altogether, these 19 multi-class datasets represent 229 binary text classification problem instances, with a positive class size of 149 on average, and class skews averaging 1:31 (median 1:17, 95th percentile 1:97).

Feature Engineering and Selection: Each feature represents the Boolean occurrence of a forced-lowercase word. Han and Karypis (2000) report having applied a stopword list and Porter's suffix-stripping algorithm. From an inspection of word counts in the data, it appears they also removed rare words that occurred < 3 times in most datasets. Stemming and stopwords were not applied to the Cora dataset, and we used the same rare word threshold. We varied the number of selected features in our experiments from 10 to 2000. Yang and Pedersen (1997) evaluated up to 16,000 words, but the F-measure had already peaked below 2000 for Chi-Squared and IG. If features are selected well, most information should be contained in the initial features selected.

4 Empirical Results

Figure 4 shows the macro-averaged F-measure for each of the feature selection metrics as we vary the number of features to select. The absolute values are not of interest here, but rather the overall trends and the separation of the top performing curves. The most striking feature is that the only metrics to perform better than using all the features available are BNS (the boldface curve) and to a limited extent IG (dashed). BNS performed best by a wide margin when using 500 to 1000 features. This is a significant result in that BNS has not been used for feature selection before, and the significance level, even in the barely visible gap between BNS and IG at 100 features, is greater than 99% confidence in a paired t-test of the 229*5 runs. Like the results of Yang and Pedersen (1997), performance begins to decline around 2000 features, and ultimately must come down to the level of using all the features.

If for scalability reasons one is limited to 20-50 features, the best available metric is IG (or Acc2, which is simpler to program). Surprisingly, Acc2, which ignores class skew, outperformed Acc, which accounts for skew. IG dominates the performance of Chi at every size of feature set.

Accuracy: The results for accuracy look qualitatively identical to those for the F-measure, although compressed into a much smaller range by the class skew (see Appendix D). BNS again performed the best by a smaller, but still >99% confident, margin. At 100 features and below, however, IG again performed best, with Acc2 being statistically indistinguishable at 20 features.

Precision-Recall Tradeoffs: As discussed, one's goal in some situations may focus on precision or on recall, rather than F-measure. The precision vs. recall scatter-plot in Figure 5 highlights the tradeoffs of the different metrics, evaluated at 1000 features selected. As with F-measure, the performance figures are macro-averaged across the five-trial average performance of each of the 229 sample problems. We see that the success of BNS with regard to its high F-measure is because it obtained on average much higher recall than any other method. If, on the other hand, precision is the central goal, IG and Chi perform best by a smaller but still significant margin to the other metrics. (At 500 features and below, IG dominated all other metrics.)

4.1 Class Skew Analysis

In classification tasks with high class skew, it can be difficult to obtain good recall, since induction algorithms are often focused on the goal of accuracy. The scatter-plot in Figure 6 shows, for each of the 229 classification tasks, its class skew vs. its average F-measure (using

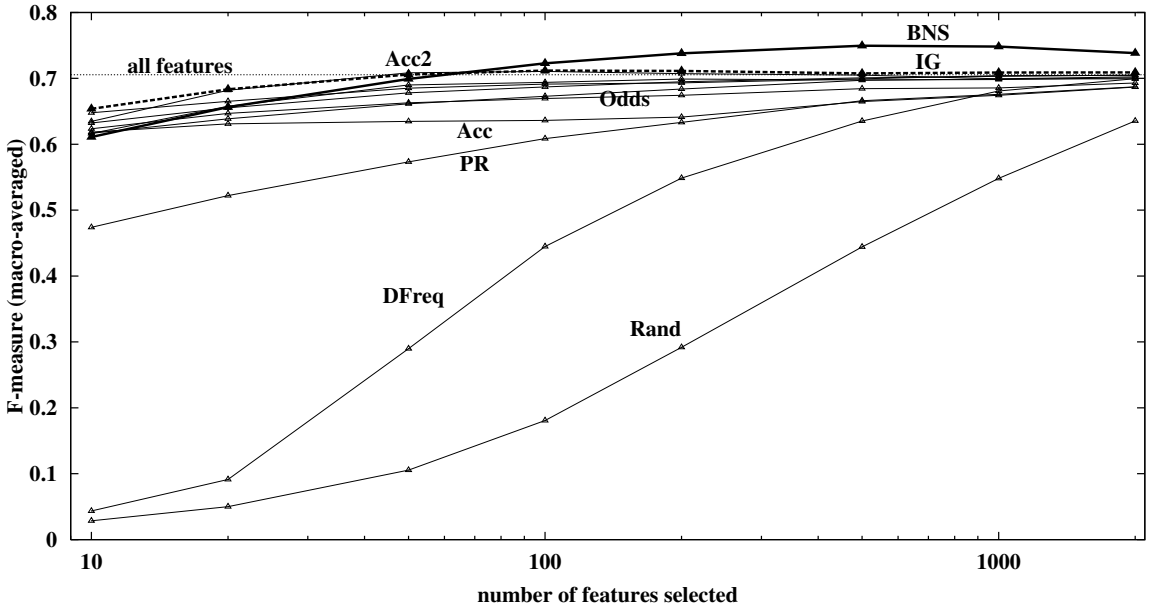


Figure 4. F-measure averaged over 229 problems for each metric, varying the number of features.

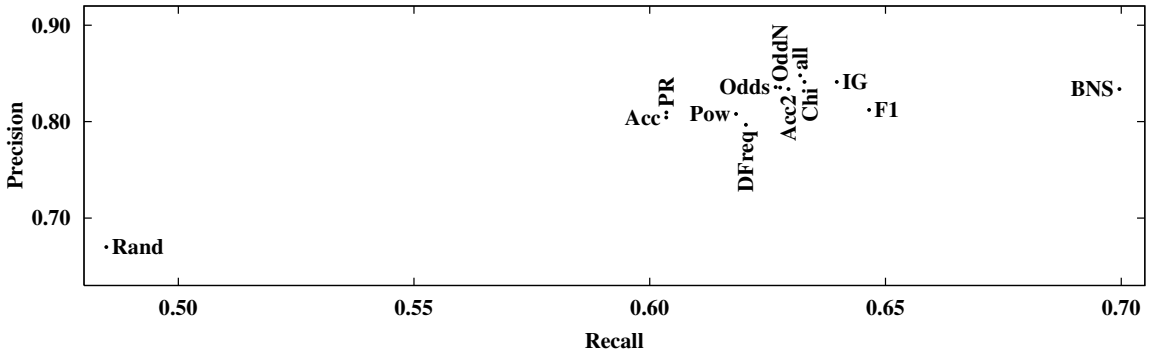


Figure 5. Precision-Recall tradeoffs from Figure 4 at 1000 features selected.

1000 features selected by BNS). We see that for lower values of class skew, the SVM classifier usually achieves good F-measure, but as the skew increases, the results vary more widely. The Cora dataset, with its controlled skew of 1:35, is visible as a vertical line of points in the figure. Although the skew is the same for each of its classes, the scores vary due to the quality of the available predictive features with respect to each class. The vertical dotted line in the figure shows the 90th percentile of class skew values studied, i.e. 23 of the 229 classification tasks have a skew exceeding 1:67. In the following analysis, we differentiate the performance of the various feature selection metrics above and below this threshold.

Figure 7 shows the F-measure performance of each of the metrics for low-skew and high-skew situations. It is remarkable that in low-skew situations, BNS is the only metric that performed substantially better than using all features. Observe that under low skew, BNS performed best overall, but if one is limited to just a few features, then IG is a much better choice. In contrast, under high skew, BNS performed best by a wide margin for any number of features selected. Figure 8 shows the same, but for Precision. Under low skew, IG performs best and eventually reaches the performance of using all features. Under high skew, on the other hand, BNS performed substantially better than IG.

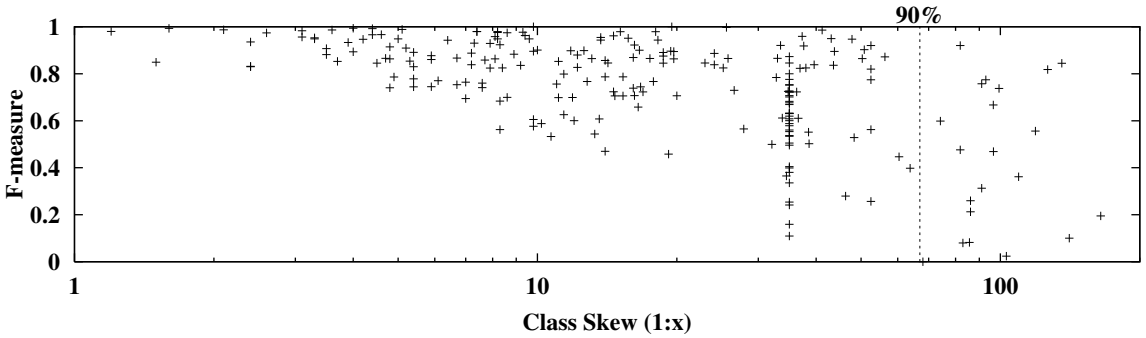


Figure 6. F-measure vs. skew for each of the 229 classification tasks. (BNS @ 1000 features)

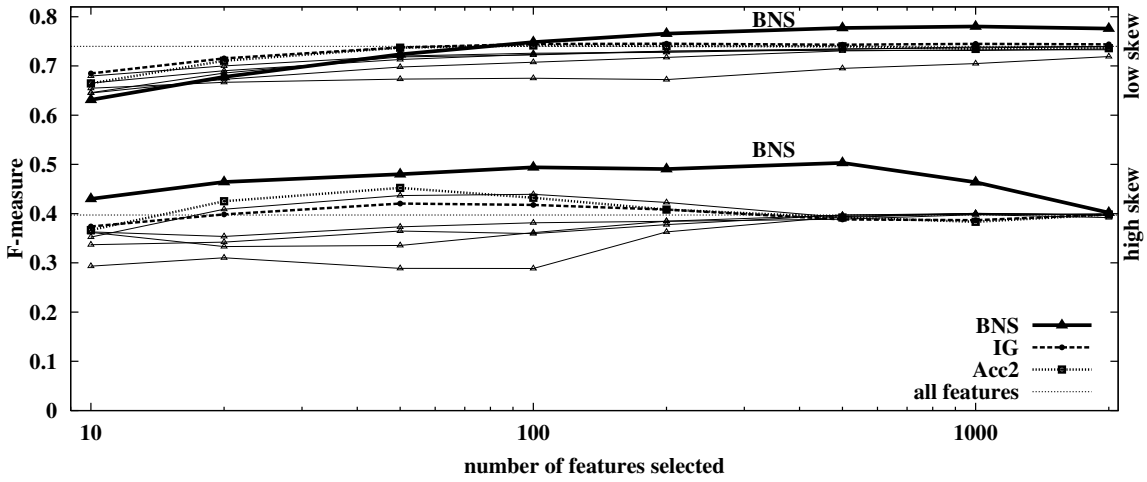


Figure 7. Average F-measure for each metric in low-skew and high-skew situations (threshold 1:67, the 90th percentile), as we vary the number of features. (To improve readability, we omitted Rand, DFreq, and PR.)

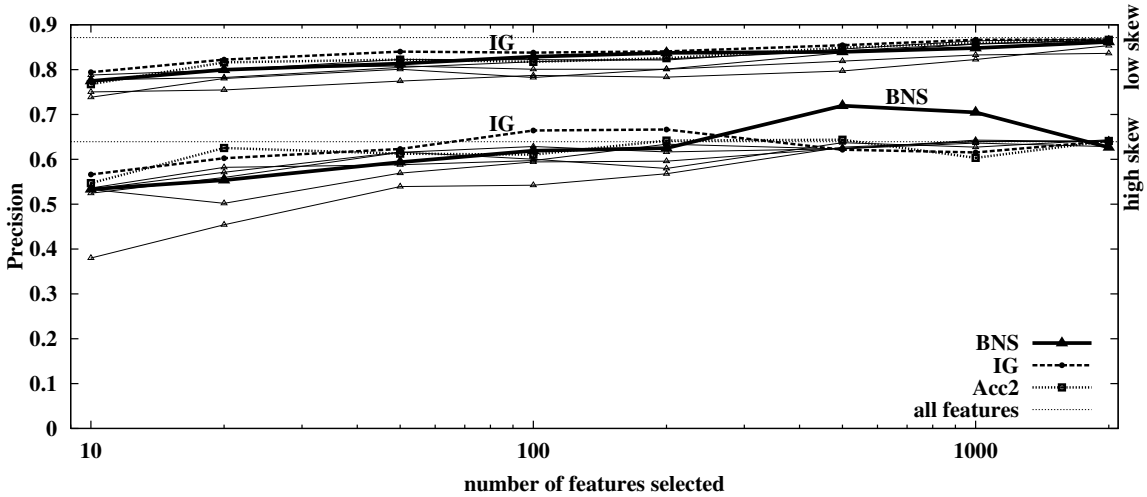


Figure 8. As Figure 7, but for precision.

4.2 Best Chances of Attaining Maximum Performance

The problem of choosing a feature selection metric is somewhat different when viewed from the perspective of a data mining practitioner whose task is to get the best performance on a *given* set of data, rather than averaging over a large number of datasets. Practitioners would like guidance as to which metric is most *likely* to yield the best performance for their single dataset at hand. Supposing the problem instance is drawn from a distribution similar to that in this study, we offer the following analysis: For each feature selection metric, we determine the percentage of the 229 problem instances for which it matched the best performance found within a small tolerance (taking the maximum over any number of features, and averaging over the 5 trials for each problem instance, i.e. 'average maximum'). We repeat this separately for F-measure, precision, recall and accuracy.

Figure 9a shows these results for the goal of maximum F-measure as we vary the acceptable tolerance from 1% to 10%. As it increases, each metric stands a greater chance of attaining close to the maximum, thus the trend. We see that BNS attained within 1% of best performance for 65% of the 229 problems, beating IG at just 40%. Figure 9b shows similar results for Accuracy, F-measure, Precision and Recall (but using 0.1% tolerance for accuracy, since large class skew compresses the range). Note that for precision, there is no single clear winner, and that several metrics beat BNS, notably IG. This is seen more clearly in Figure 10a, which shows these results for varying tolerances. IG consistently dominates at higher tolerances, though the margin is less striking than for BNS in Figure 9a. (Rand is below 50%.)

4.2.1 Residual Win Analysis

If one were willing to invest the extra effort to try two different metrics for one's dataset at hand and select the metric with better precision via cross-validation, the two leading metrics, IG and Chi, would seem a logical choice. (Referring to Figure 5 and Figure 10a.) However, it may be that whenever IG fails to attain the maximum, Chi also fails. To evaluate this, we performed a residual analysis for each metric in which we counted the residual problem instances where it attained near optimum *on only those tasks for which the leading metric failed.* Indeed, whenever IG failed to attain the maximum, Chi had the most correlated failures and was as bad as Rand. When IG failed, BNS performed the best, which is surprising given its lackluster precision seen in Figure 10a. Figure 10b shows these results represented as the total percentage of problems for which IG *or* a second 'backup' metric attained the best precision. In contrast to Chi, BNS had the least correlated failures with IG and so it is a better backup choice.

This led us to repeat the analysis to determine the optimal pair of metrics that together attained the best precision most often. The best pair found for precision is BNS+Odds (overlaid on Figure 10b as a dot-dashed line). It is surprising to some extent that the top individual metric, IG, is not a member of the best pair. We repeated this analysis for each goal: For recall, BNS+F1 was best by a very wide margin compared to other pairs. Less strikingly, BNS+IG was best for F-measure, and BNS+OddN was best for accuracy.

5 Discussion

It is difficult to beat the performance of SVM using all available features. In fact, it is sometimes claimed that feature selection is unnecessary for SVMs. Nonetheless, this study shows that BNS can improve the performance for all goals (except precision in low-skew situations, where using all the features may give better results).

Of the other metrics, IG is often competitive. IG is used by decision tree algorithms, e.g. C4.5, to select on which feature each node should split. Our findings raise the possibility that BNS may be useful to enhance decision trees for nodes with high class skew.

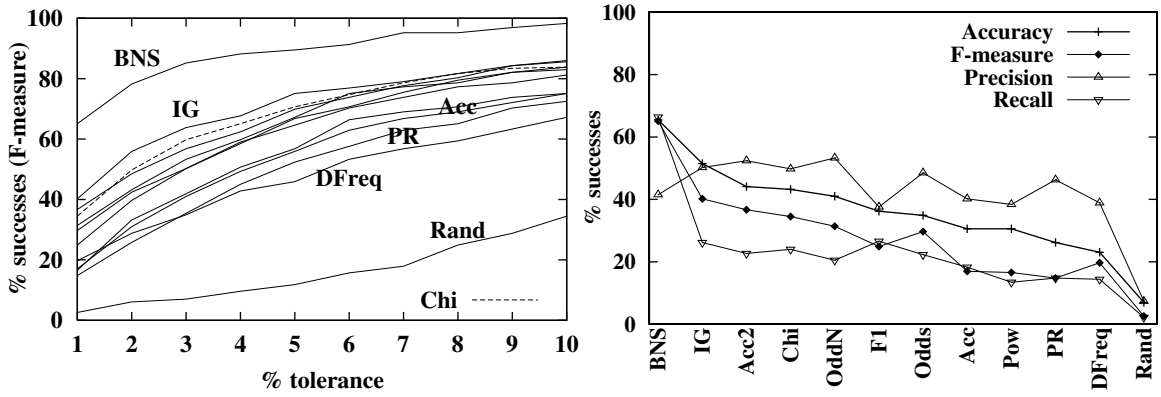


Figure 9. (a) Percentage of problems on which each metric scored within x% tolerance of the best F-measure of any metric. (b) Same, for F-measure, recall, and precision at a fixed tolerance of 1%, and for accuracy at a tolerance of 0.1%.

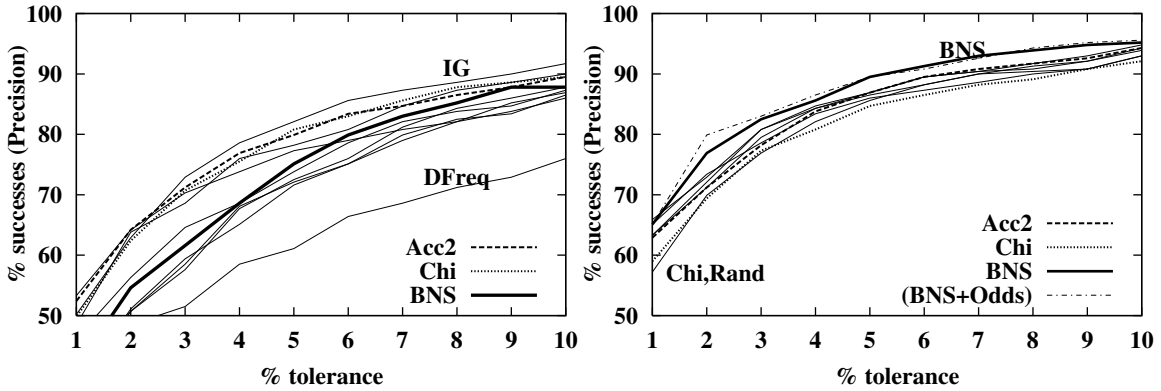


Figure 10. (a) As Figure 9a, but for precision. (b) Same axes and scale, but for each metric combined with IG. (Except the BNS+Odds curve is not combined with IG.)

We have performed this study with binary features. For real-valued attributes, one may either threshold them to produce binary features, or replace the *tp* and *fp* counts with scaled sums. It would be useful to generalize BNS somehow for nominal valued attributes.

In the foregoing experiments, positively and negatively correlated features were selected symmetrically. To better understand the role that each plays in skewed problems, we performed a similar suite of experiments on the Cora dataset where we systematically varied their relative weighting. We multiplied the BNS score by a weight α for positive features, and by $(1-\alpha)$ for negative features. Refer to the precision and recall graphs in Figure 22 in Appendix D. Briefly, they reveal that by preferring positive features (α=60%), we get good precision with relatively few features selected, however, recall suffers in comparison with unweighted BNS. If we instead prefer negative features (α=40%), we can get outstanding recall once many features are selected, but at the cost of poor precision. If we nearly eliminate positive features (α=10%), we defeat the induction algorithm altogether. We conclude that the role of positive features is precision (which is a widely held belief), and the role of negative features is recall. Optimal F-measure on the entire benchmark was obtained with the two in balance, i.e. unweighted BNS. Given this understanding and the graphical analysis presented in Figure 2, we can better appreciate why unbalanced feature selection metrics such as PR, DFreq and Chi are not effective.

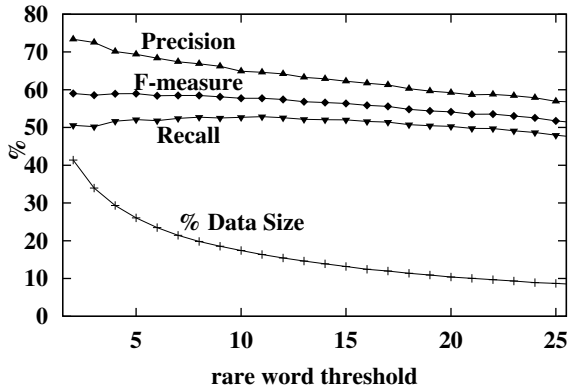


Figure 11. Performance for BNS at 1000 features as we vary the rare-word cutoff.

In the preparation of the datasets, word features were omitted that occurred fewer than two times in the (training & testing) corpus. Some text preparation practices remove many more rare words to dramatically reduce the size of the data. We performed an experiment with the Cora dataset, varying this rare word threshold up to 25 occurrences. At each threshold, we measured the F-measure, precision, recall and accuracy (averaged over five runs and macro-averaged over all 36 classes) for SVM using 1000 features selected via BNS. As shown in Figure 11, each of these performance measures tends to drop as the threshold increases. (The accuracy curve has a linear decrease from 98.2% to 97.5%, but is omitted to improve readability of the other curves.) As an exception to the trend, recall first experiences a slight rise up to a threshold of about ten occurrences. This is explainable given our understanding of the roles of positive and negative features, and the fact that the rare word threshold removes words near the origin. The effect is to remove some eligible positive words, forcing BNS to select more negative words, which aids recall at the cost of precision.

We conclude that to effectively reduce the size of one's dataset without adversely impacting classification performance, one should set the rare word cutoff low, and then perform aggressive feature selection using a metric (which runs in linear time to the size of the dataset). If recall is one's sole goal, then a greater proportion of rare words should be eliminated.

6 Conclusion

This paper presented an extensive comparative study of feature selection metrics for the high-dimensional domain of text classification, focusing on support vector machines and 2-class problems, typically with high class skew. It revealed the surprising performance of a new feature selection metric, Bi-Normal Separation.

Another contribution of this paper is a novel evaluation methodology that considers the common problem of trying to select one or two metrics that have the best chances of obtaining the best performance for a *given* dataset. Somewhat surprisingly, selecting the two best-performing metrics can be sub-optimal: when the best metric fails, the other may have correlated failures, as is the case for IG and Chi for maximizing precision. The residual analysis determined that BNS paired with Odds Ratio yielded the best chances of attaining the best precision. For optimizing recall, BNS paired with F1 was consistently the best pair by a wide margin.

Future work could include extending the results for nominal and real-valued feature values, and demonstrating BNS for non-text domains. The feature scoring methods we considered are oblivious to the correlation between features; if there were ten duplicates of a predictive feature, each copy would be selected. To handle this, wrapper techniques are called for, which search for an optimal *subset* of features (Kohavi & John, 1997; Guyon et al., 2002). BNS may prove a

useful heuristic to guide such a search or to perform pre-selection of features for increased scalability. Finally, recent research has shown that tuning parameters such as C and B for SVMs may yield significant performance improvements. As feature selection and model tuning have been studied independently, there lies an opportunity to study the interaction of the two together.

Acknowledgments

We would like to thank the anonymous reviewers for their time and helpful feedback. We also extend our thanks to the WEKA project for their open-source machine learning software (Witten 1999), and to Han & Karypis (2000) and Tom Fawcett for the prepared datasets. We greatly appreciate Jaap Suermondt's input, Hsiu-Khuern Tang's competent and willing statistics consulting, and the aid of Fereydoon Safai, Ren Wu, Mei-Siang Chan, and Richard Bruno in securing computing resources. We also thank the Informatics and Distribution Laboratory (http://www-id.imag.fr/grappes) for use of the ID/HP i-cluster.

Note: The datasets and additional color graphs of results can be found in the online appendices at
http://www.jmlr.org/papers/volume3/forman03a/abstract.html
http://www.hpl.hp.com/techreports/2002/

Appendix A: Datasets

Dataset	Source	Docs	Words	Ratio	Cutoff	Classes	Class Sizes (sorted)
cora36	whizbang.com	1800	5171	3	3	36	50 (each)
fbis	TREC	2463	2000	1	10	17	38 43 46 46 46 48 65 92 94 119 121 125 139 190 358 387 506
la1	TREC	3204	31472	10	1	6	273 341 354 555 738 943
la2	TREC	3075	31472	10	1	6	248 301 375 487 759 905
oh0	OHSUMED	1003	3182	3	3	10	51 56 57 66 71 76 115 136 181 194
oh5	OHSUMED	918	3012	3	3	10	59 61 61 72 74 85 93 120 144 149
oh10	OHSUMED	1050	3238	3	3	10	52 60 61 70 87 116 126 148 165 165
oh15	OHSUMED	913	3100	3	3	10	53 56 56 66 69 98 98 106 154 157
ohscal	OHSUMED	11162	11465	1	3	10	709 764 864 1001 1037 1159 1260 1297 1450 1621
re0	Reuters-21578	1504	2886	2	3	13	11 15 16 20 37 38 39 42 60 80 219 319 608
re1	Reuters-21578	1657	3758	2	3	25	10 13 15 17 18 18 19 19 20 20 27 31 31 32 37 42 48 50 60 87 99 106 137 330 371
tr11	TREC	414	6429	16	3	9	6 11 20 21 29 52 69 74 132
tr12	TREC	313	5804	19	3	8	9 29 29 30 34 35 54 93
tr21	TREC	336	7902	24	3	6	4 9 16 35 41 231
tr23	TREC	204	5832	29	3	6	6 11 15 36 45 91
tr31	TREC	927	10128	11	3	7	2 21 63 111 151 227 352
tr41	TREC	878	7454	8	3	10	9 18 26 33 35 83 95 162 174 243
tr45	TREC	690	8261	12	3	10	14 18 36 47 63 67 75 82 128 160
wap	WebACE	1560	8460	5	3	20	5 11 13 15 18 33 35 37 40 44 54 65 76 91 91 97 130 168 196 341

The 'ratio' is the number of words divided by the number of documents, and
is directly influenced by the rare-word cutoff used, shown in the following column.

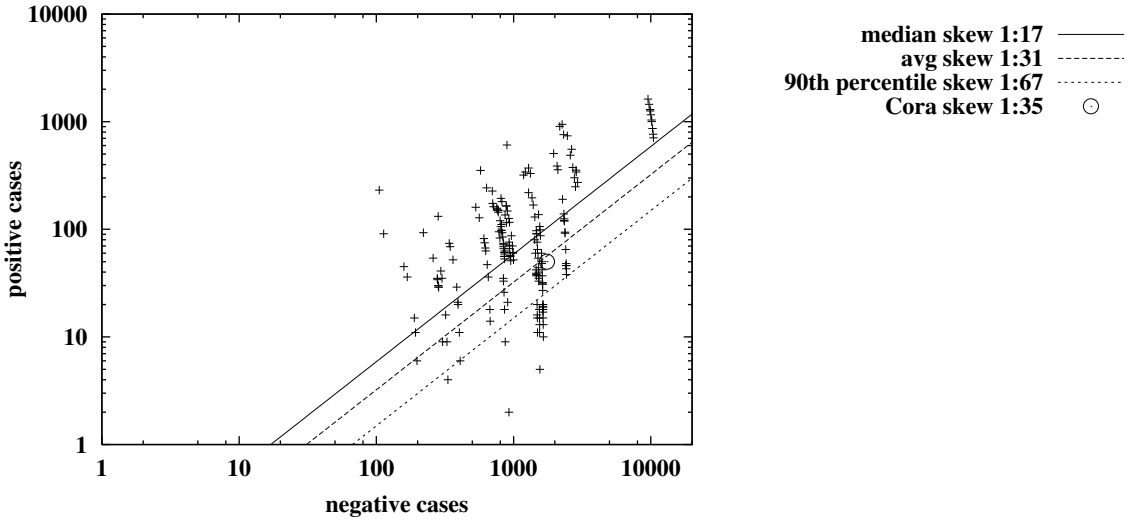


Figure 12. Sizes of positive and negative classes for each of the 229 binary classification tasks. Note the Cora dataset has 36 data points overlaid at (1750,35).

References

Susan Dumais, John Platt, David Heckerman and Mehran Sahami. Inductive Learning Algorithms and Representations for Text Categorization. In *Proceedings of the 17th International Conference on Information and Knowledge Management*, pages 148-155, Maryland, 1998.

Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning,* 46(1-3):389-422, 2002.

Eui-Hong Sam Han and George Karypis. Centroid-Based Document Classification: Analysis & Experimental Results. In *Proceedings of the Fourth European Conference on the Principles of Data Mining and Knowledge Discovery* (PKDD), pages 424-431, Lyon, France, 2000.

James A. Hanley. The Robustness of the "Binormal" Assumptions Used in Fitting ROC Curves. *Medical Decision Making*, 8(3):197-203, 1988.

Thorsten Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proceedings of the Tenth European Conference on Machine Learning* (ECML), pages 137-142, Berlin, Germany, 1998.

Ron Kohavi and George H. John. Wrappers for Feature Subset Selection. *Artificial Intelligence*, 97(1-2):273-324, 1997.

George A. Miller and Edwin B. Newman. Tests of a statistical explanation of the rank-frequency relation for words in written English. *American Journal of Psychology*, 71:209-218, 1958.

Andrew McCallum and Kamal Nigam. A Comparison of Event Models for Naive Bayes Text Classification. Workshop on Learning for Text Categorization, In the *Fifteenth National Conference on Artificial Intelligence* (AAAI), 1998.

Dunja Mladenic and Marko Grobelnik. Feature Selection for Unbalanced Class Distribution and Naïve Bayes. In *Proceedings of the Sixteenth International Conference on Machine Learning* (ICML), pages 258-267, 1999.

Bernhard Schölkopf and Alex Smola. *Learning with Kernels.* MIT Press, Cambridge, MA, 2002.

Adrian J. Simpson and Mike J. Fitter. What is the Best Index of Detectability? *Psychological Bulletin*, 80(6):481-488, 1973.

Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999.

Yiming Yang and Xin Liu. A Re-examination of Text Categorization Methods. In *Proceedings of the Twenty-Second International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR), pages 42-49, 1999.

Yiming Yang and Jan O. Pedersen. A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the 14th International Conference on Machine Learning* (ICML), pages 412-420, 1997.

Appendix B: Experimental Procedure

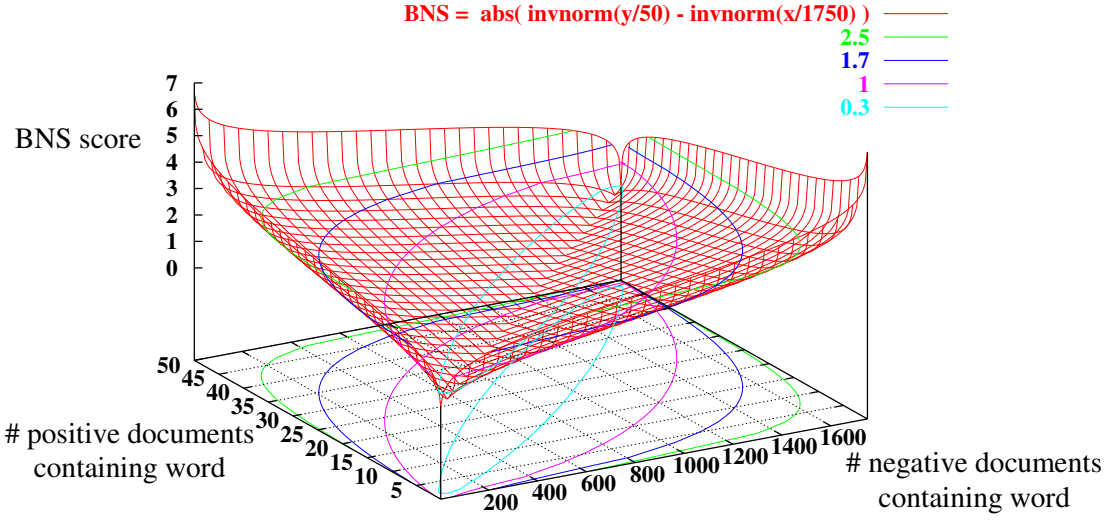
The following pseudo-code details the experimental procedure for collecting and processing the data:

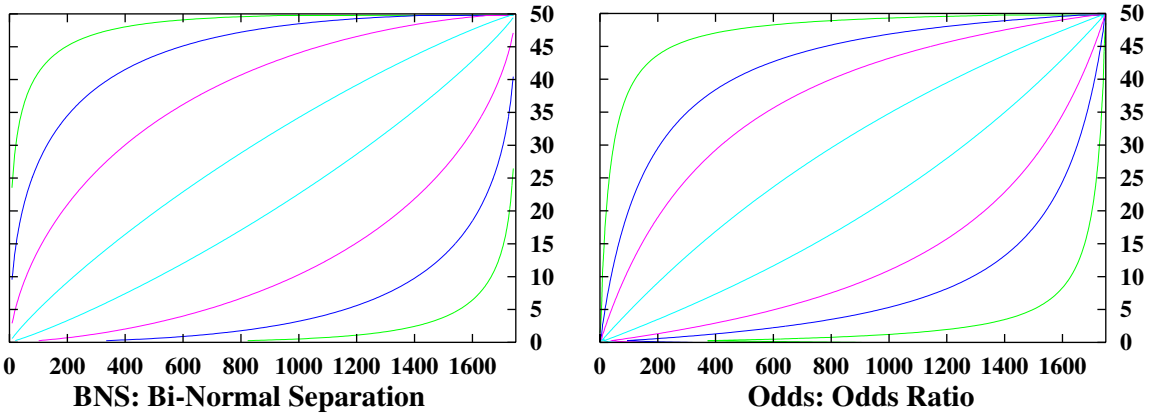
```
for each dataset d:
|  for each class c of the dataset d, using c as the positive class and the others as the negative class:
|     for N=5 random trials:
|     |  for each random splits of the dataset for 4-fold stratified cross-validation:
|     |     for each feature selection metric:
|     |     |  for each number of features to select—10,20,50,100,200,500,1000,2000:
|     |     |     select top features via the metric using only the training set
|     |     |     train a SVM classifier on the training set split
|     |     |     measure performance on the testing set split
|     |     |  end
|     |     end
|     |  end
|     |  record the 4-fold cross-validation scores for  accuracy, precision, recall, F-measure
|     |  (also record the maximum attained over any number of features)
|     end
|     determine the average scores over all N=5 trials
|     record these under the unique key: dataset, class, feature selection metric, number of features
|     (record the 'average maximum' attained with key: dataset, class, feature selection metric)
|  end
end
macro-average the performance measures over all 229 classification tasks
record the results under the key: feature selection metric, number of features
for each goal—accuracy, precision, recall, F-measure:
|  for each of the 229 tasks:
|     determine the best 'average maximum' performance attained by any metric & number of features
|  end
|  for each tolerance level t from 1% to 10%:
|     for each feature selection metric:
|     |  determine percentage of problems on which the metric ties for best within t% tolerance
|     end
|  end
end
```

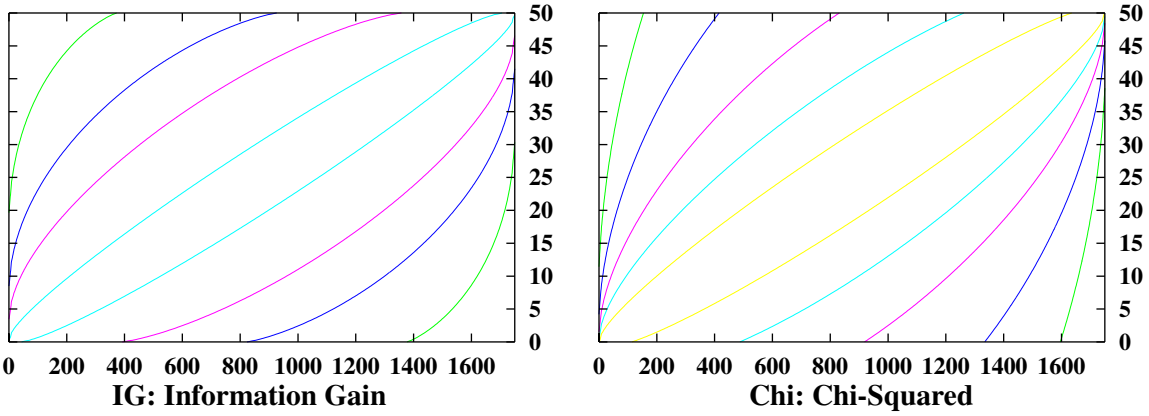
This resulted in nearly half a million invocations of the learning algorithm, run on over a hundred different machines. We performed additional variations of this procedure for several side studies.

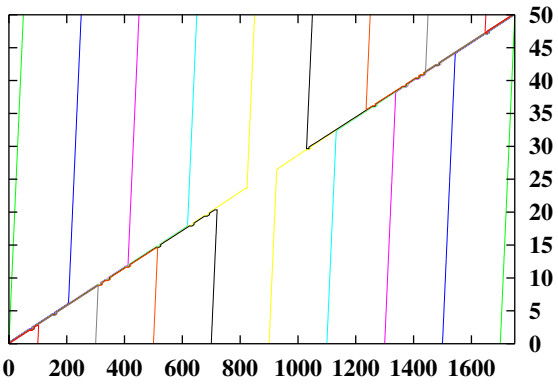
Appendix C: Graphical Analysis of Feature Selection Metrics

This section contains color graphs illustrating the shape of each feature selection metric. Each graph shows a set of isocline contours, as a topographic map. For reference, compare the BNS isoclines to its three-dimensional plot below, which includes contours both on the surface and projected down on the plane below. Like colored isoclines indicate the same value within a plot (but are not comparable between plots). Any feature in the top left or bottom right corners would be a perfect predictor, and so each metric scores highest in these corners, except DFreq, which gives highest value to the most frequent features, i.e. the top right.

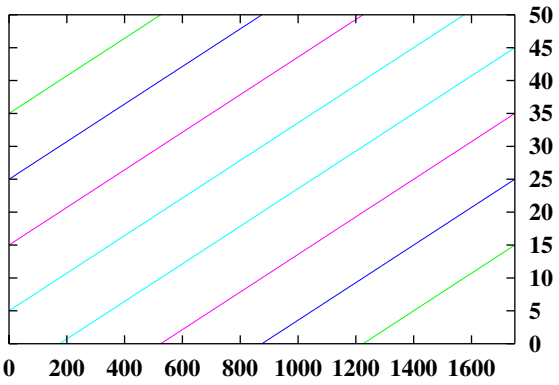




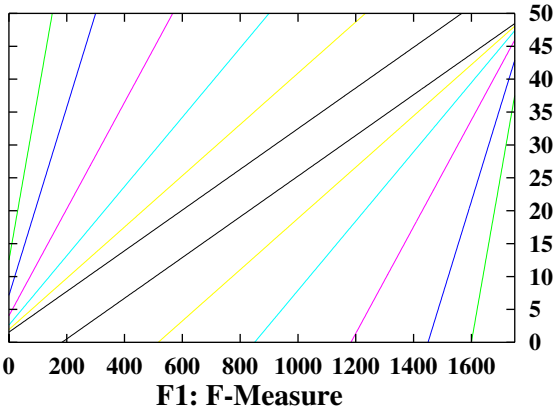




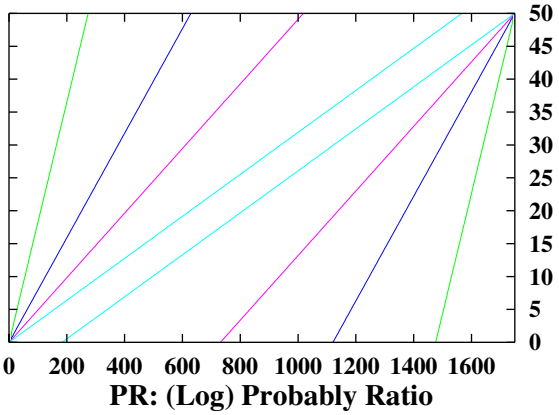
Acc: Accuracy



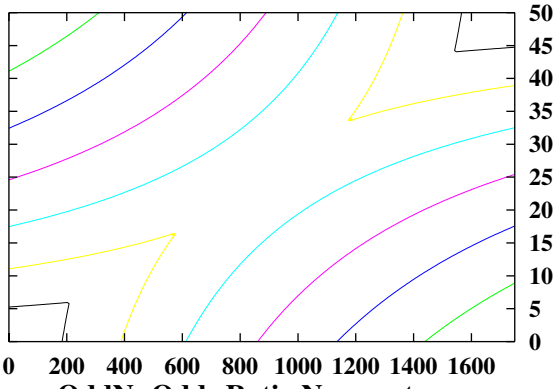
Acc2: Accuracy (ignoring skew)



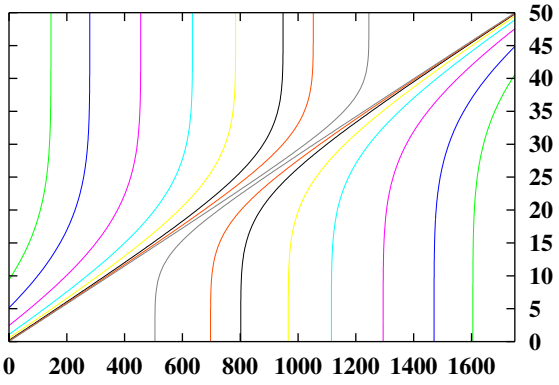
F1: F-Measure



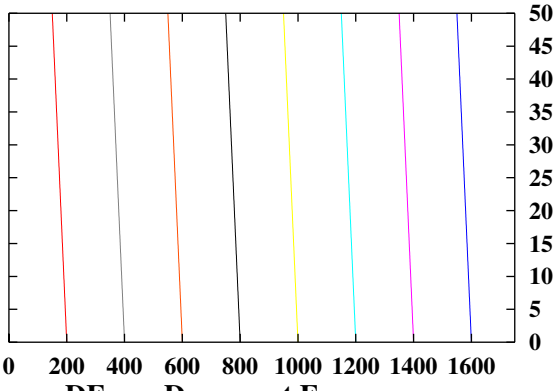
PR: (Log) Probably Ratio



OddN: Odds Ratio Numerator



Pow: Power



DFreq: Document Frequency

Rand: Random (varies)

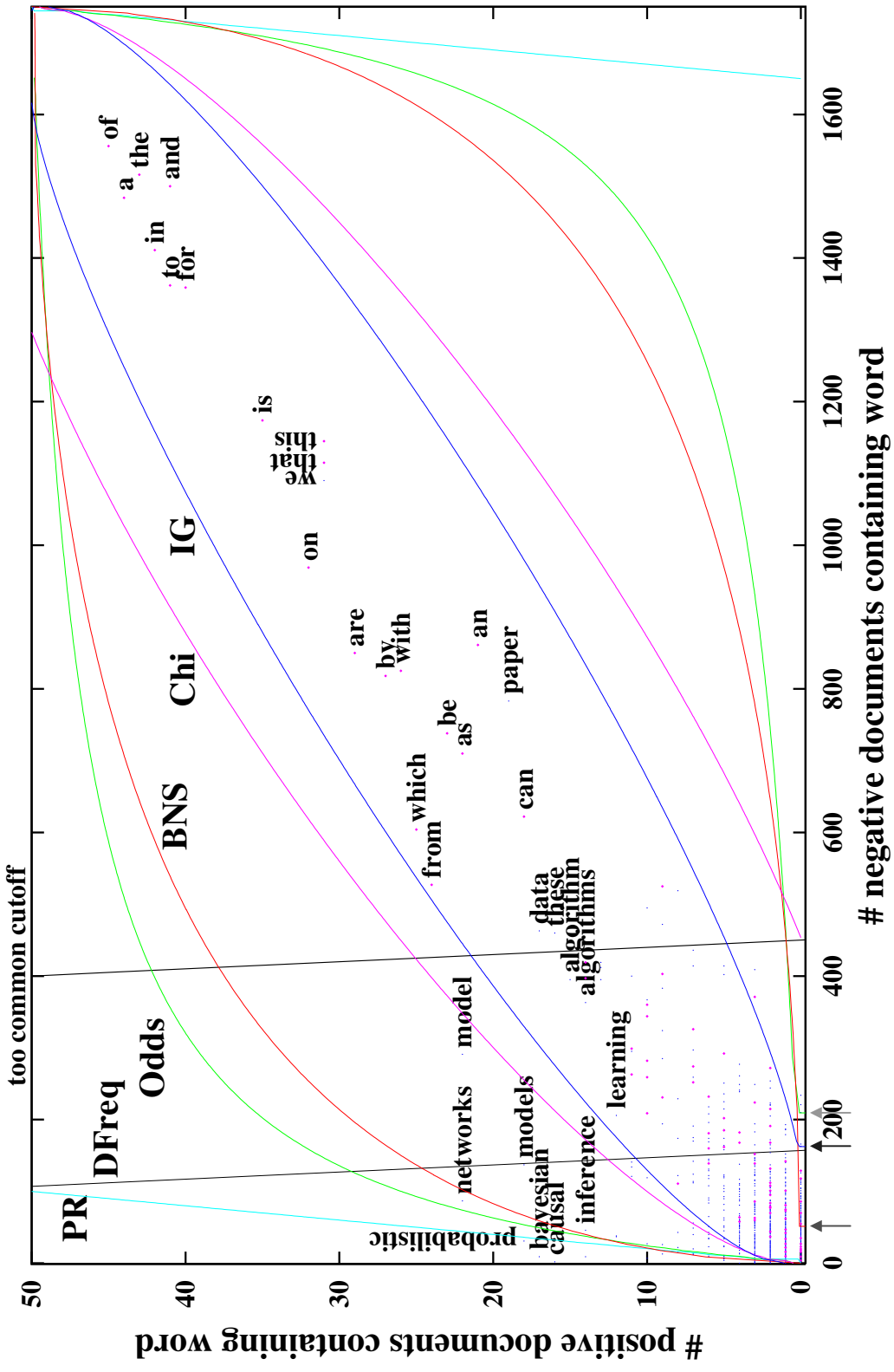


Figure 13. Color version of Figure 2. Note that BNS selects many more negative features.

Appendix D: Additional Color Graphs of Results

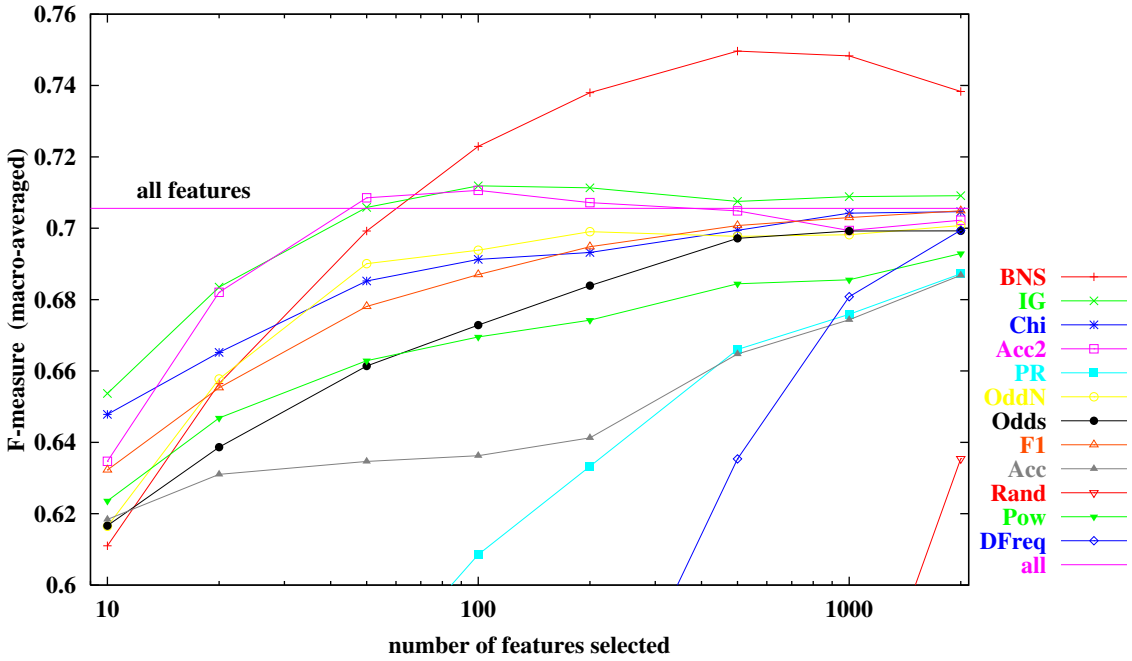


Figure 14. Color zoomed version of Figure 4. F-measure macro-averaged over repeated trials on 229 binary text classification tasks, as we vary the number of features selected for each feature selection metric.

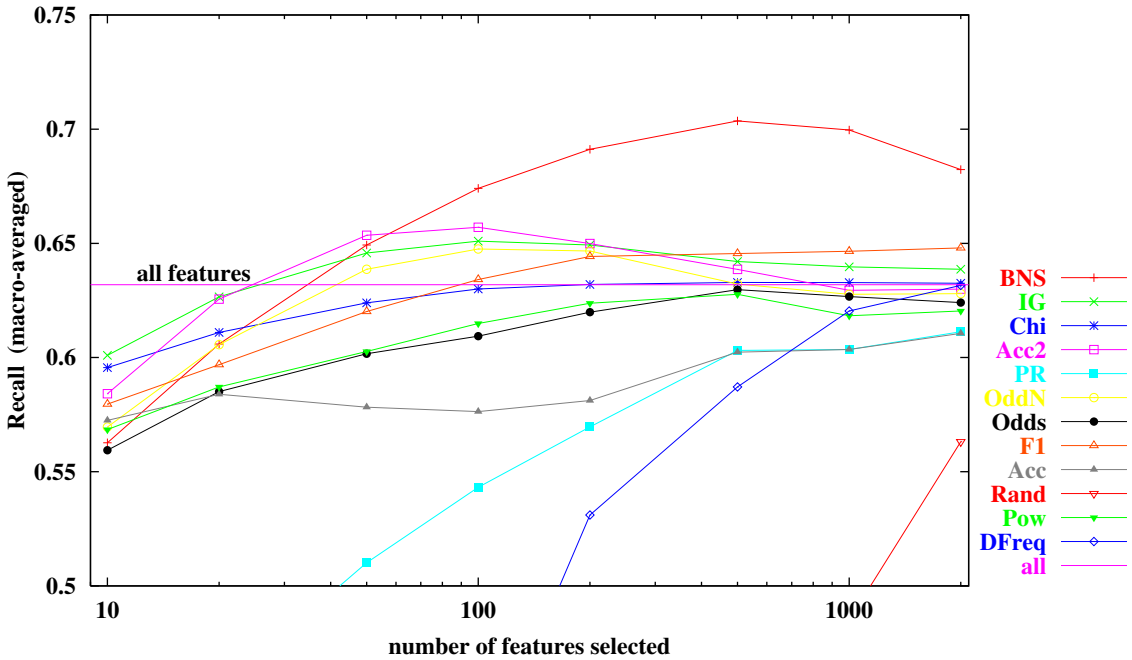


Figure 15. Same as Figure 14, but for recall.

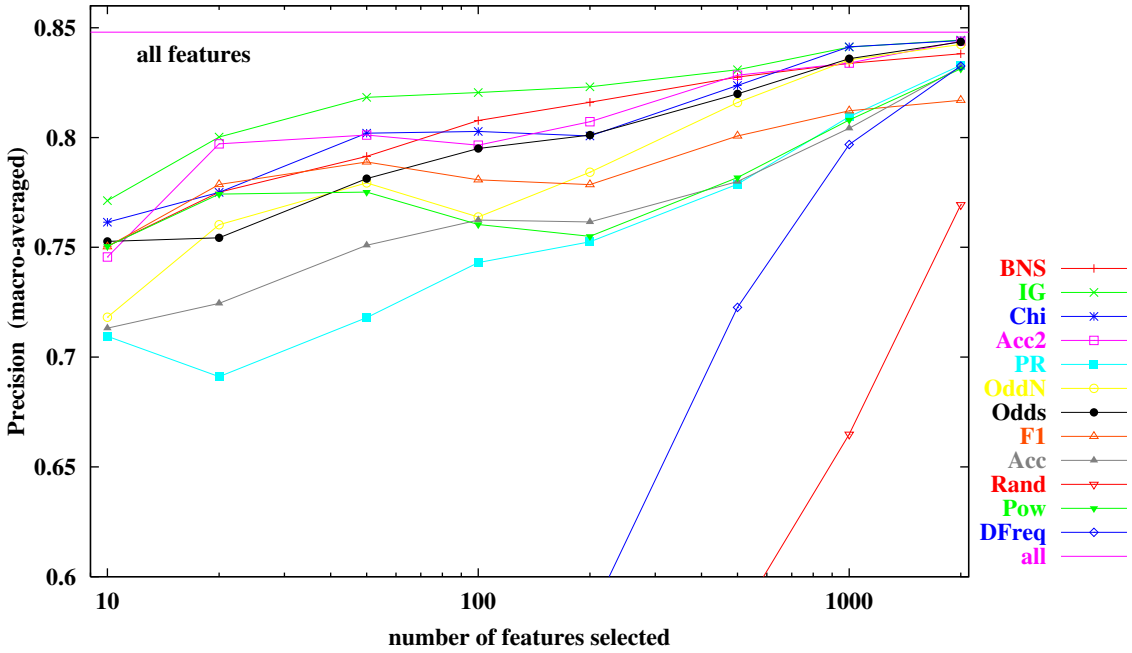


Figure 16. Same as Figure 14, but for precision.

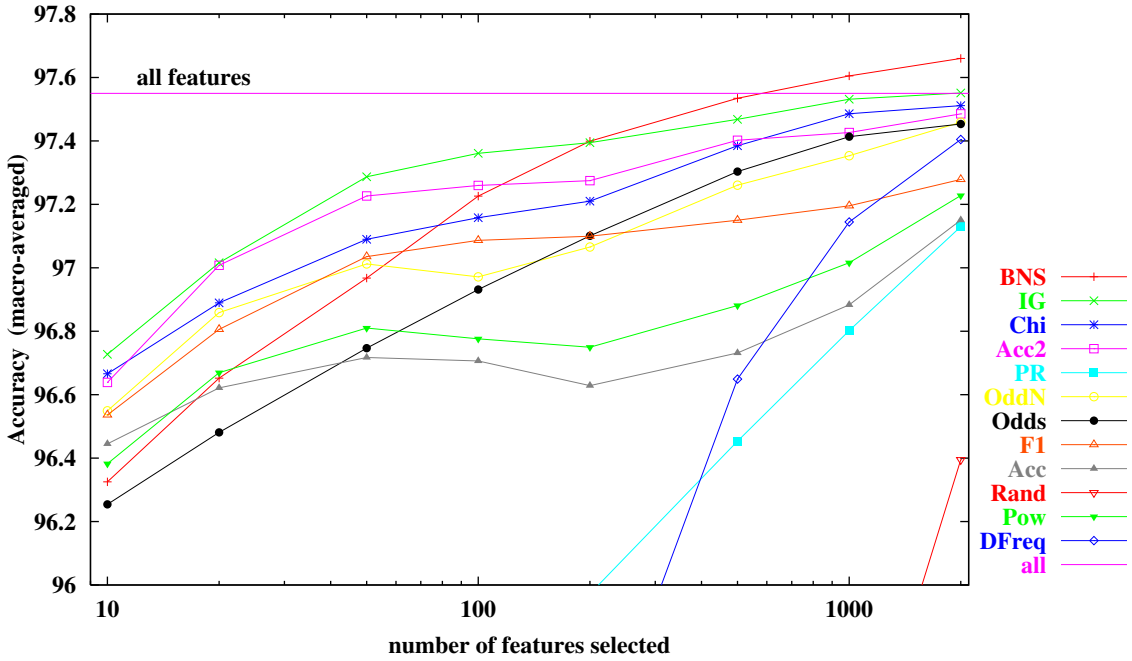


Figure 17. Same as Figure 14, but for accuracy.

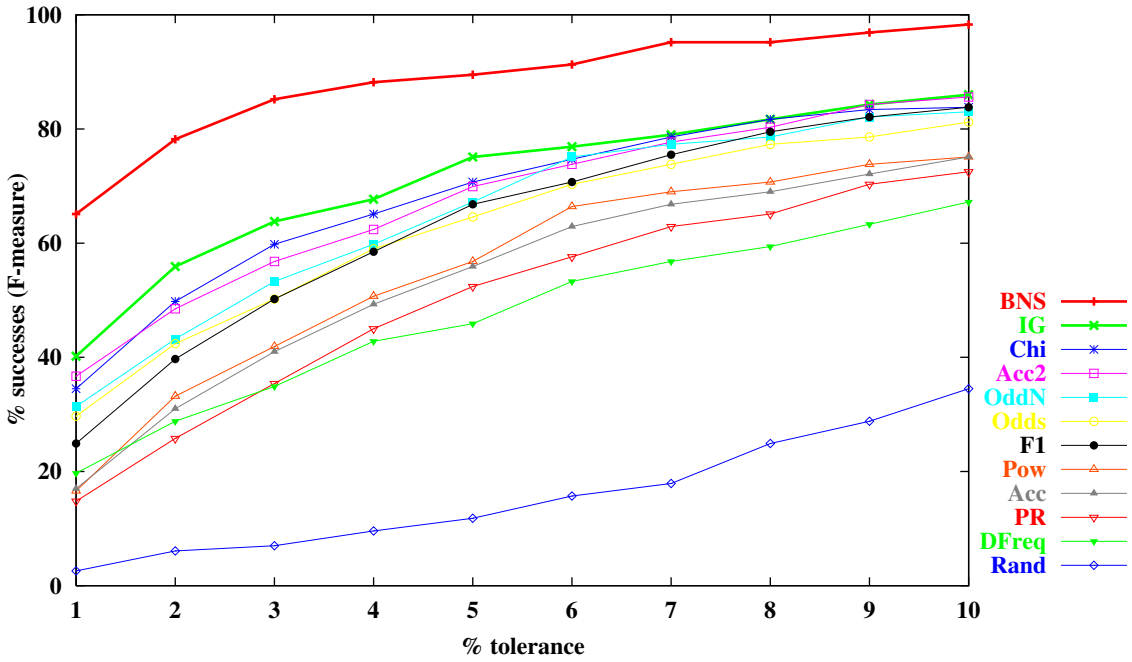


Figure 18. Color version of Figure 9a. Percentage of problems on which each metric scored within x% tolerance of the best F-measure achieved by any metric.

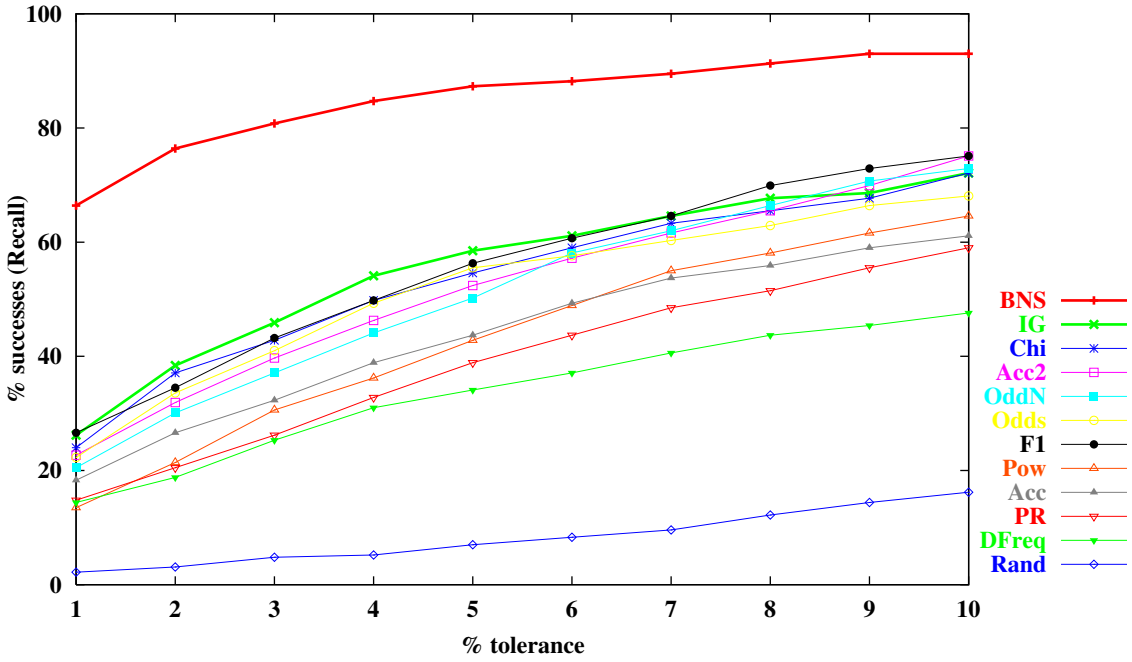


Figure 19. Same as Figure 18, but for recall.

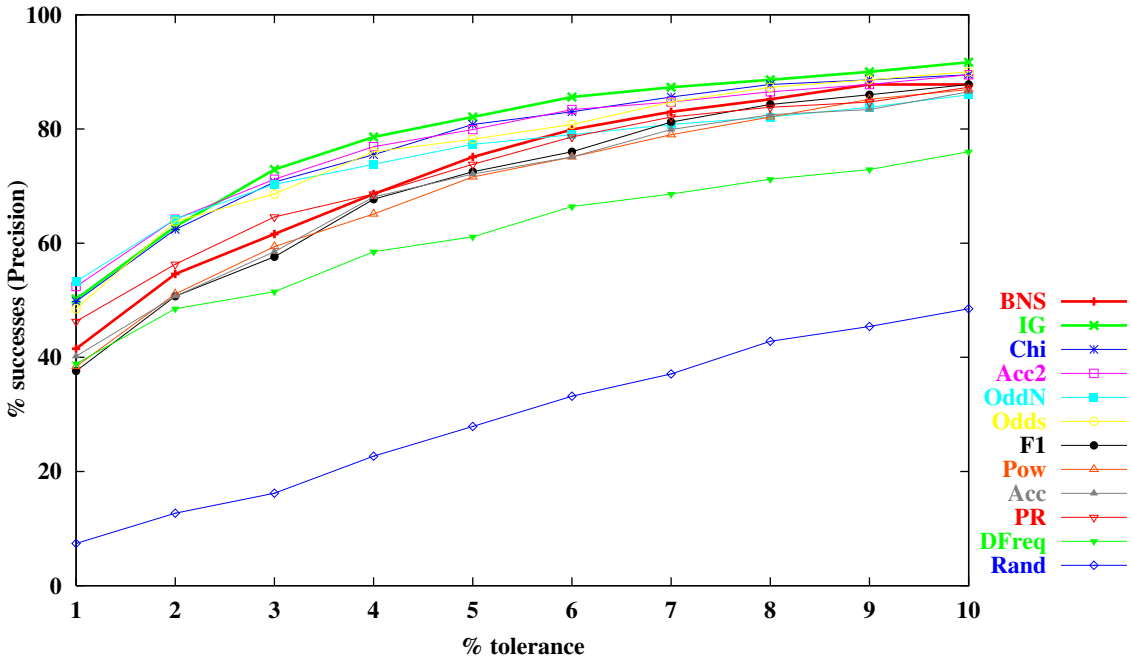


Figure 20. Same as Figure 18, but for precision.

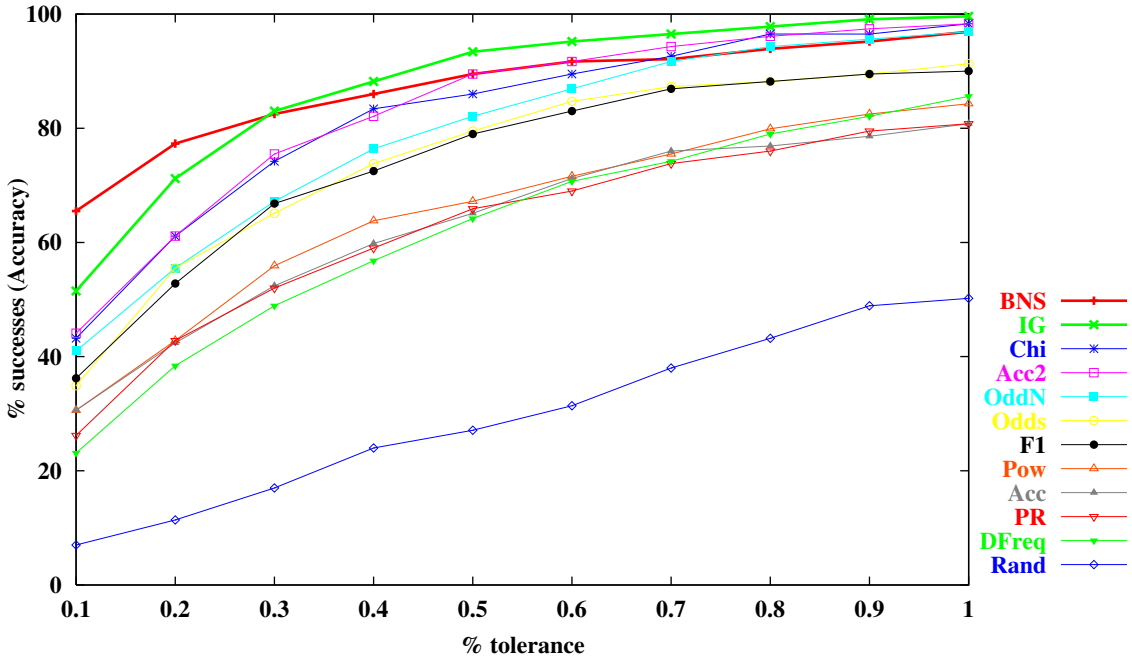


Figure 21. Same as Figure 18, but for accuracy. Note the smaller x-axis scale.

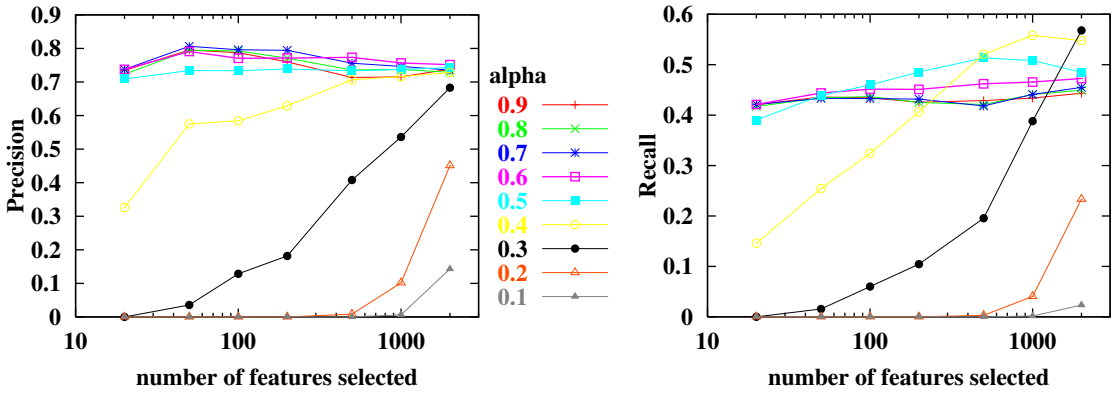


Figure 22. (a) Precision vs. number of features selected, as we systematically vary the preference weight α for positive vs. negative features in BNS. High α prefers positive features. (b) Same, but for Recall. (Note: only the Cora dataset was used, so the results are not comparable to other figures. While α=40% yields a superior F-measure for the 36 classes of the Cora dataset on average, for the entire benchmark of 229 tasks, we were unable to beat the F-measure of unweighted BNS by experimenting with different values of α.)