A Pitfall and Solution in Multi-Class Feature Selection for Text Classification

George Forman GHFORMAN@HPL.HP.COM
Hewlett-Packard Labs, 1501 Page Mill Road, Palo Alto, CA 94304 USA

Abstract

Information Gain is a well-known and empirically proven method for high-dimensional feature selection. We found that it and other existing methods failed to produce good results on an industrial text classification problem. On investigating the root cause, we find that a large class of feature scoring methods suffers a pitfall: they can be blinded by a surplus of strongly predictive features for some classes, while largely ignoring features needed to discriminate difficult classes. In this paper we demonstrate this pitfall hurts performance *even* for a relatively uniform text classification task. Based on this understanding, we present solutions inspired by round-robin scheduling that avoid this pitfall, without resorting to costly wrapper methods. Empirical evaluation on 19 datasets shows substantial improvements.

1. Introduction

A customer support division of Hewlett-Packard approached our data-mining department in HP Labs with an apparently straightforward text classification task: sorting millions of technical support documents into topic categories. In the process of developing a machine learning solution, we found that well-established feature selection methods failed to perform tolerably.

Our in-depth study of the problem revealed that there is a remarkably pervasive pitfall in most multi-class feature scoring methods, such as Information Gain, Mutual Information and Chi-Squared. It arises in situations where one or a few 'easy' classes have many strongly predictive features, and other 'difficult' classes have only weakly predictive features. All feature selection methods that evaluate features independently will be consistently drawn to the many strong predictors of the easy classes, and will be distracted from selecting features to help discriminate the difficult classes. In the extreme, a tremendous surplus of excellent predictive features for one class can effectively hide all useful features for discerning other classes. This is not far fetched: for example, consider classifying email into folders, where one particular folder represents a very dissimilar class, e.g. German or spam.

We encountered this problem—which we call the 'siren pitfall'—in the technical support classification task, and we speculate that it may be more common in practical industrial tasks than in the typically homogeneous benchmark datasets often studied in the research literature. Nonetheless, this study shows the issue is pervasive even in homogeneous tasks.

In Section 2 we analyze and illustrate the pitfall for an example dataset. Rather than exhibit it for our industrial task—which may leave the reader unconvinced as to how frequently the pitfall may occur in practice—we demonstrate that it occurs *even with a well balanced, homogeneous task*: a dataset of research paper abstracts in 36 categories representing different fields of computer science with exactly 50 cases for each class.

Given this understanding, we then present in Section 3 a family of feature selection algorithms, motivated by round-robin allocation. They avoid the risks of completely independent feature evaluation, and at the same time avoid compute-intensive wrapper methods. In Section 4 we evaluate the performance improvement on a set of 19 multi-class text classification tasks. We summarize and suggest future work in Section 5. The balance of the introduction further defines the scope of this work, and contrasts with related work.

1.1 Scope and Related Work

The scope of this paper implicates all feature scoring methods that evaluate features independently. This excludes wrapper methods, which apply general search mechanisms, such as sequential forward selection or genetic search, with repeated calls to the induction algorithm subroutine to evaluate various subsets of features (Hall & Holmes, 2003; Kohavi & John, 1997). Wrapper methods effectively consider the joint distribution of features—to the extent that the underlying induction algorithm can. However, they involve great computational cost and are inappropriate for high dimensional tasks. In text classification, the number of potential word features commonly exceeds the number of training examples by more than an order of magnitude,

Appearing in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004. Copyright 2004 by Hewlett-Packard.

not to mention the explosive number of potential word phrase features (Mladenic & Grobelnik, 1998). In contrast, feature-scoring methods, such as Information Gain, run much faster because they evaluate features independently without considering feature interactions and without induction.

There has been a number of feature selection studies applied to binary (two-class) classification tasks (e.g. Forman, 2003; Guyon, Weston, Barnhill & Vapnik, 2002; Mladenic & Grobelnik, 1999). This paper, however, focuses on *multi-class* or *1-of-M* tasks: choosing one of M nominal classes, where M>2. Applications include filing documents in folders, or routing undirected customer email to the most appropriate department.

Some feature selection studies have been performed in the framework of *topic identification* or *N-of-M* tasks (e.g. Yang & Pedersen, 1997). This formulation comprises a set of M independent binary classification tasks. The solution to such problems is to provide each of the M binary classifiers with its own optimized feature selection. In some situations, however, it may be necessary due to system constraints to select one set of features to be used by all binary classifiers; in this case, this paper applies.

2. Pitfall in All Scoring Methods

Feature scoring methods consider each feature independently, examining the counts in the contingency table across classes. Mainstream scoring methods include Information Gain (IG), Mutual Information, Chi-Squared (CHI), and variations on Term Frequency x Inverse Document Frequency (Mladenic et al., 1999). Yang and Pedersen (1997) performed a controlled study in the text classification domain and found CHI and IG to be top performers. What is common to all of these feature-scoring methods is that they conclude by ranking the features by their independently determined scores, and then select the top scoring features.

The level of difficulty of text classification tasks naturally varies. As the number of distinct classes increases, so does the difficulty, and therefore the size of the training set needed. In any multi-class text classification task, inevitably some classes will be more difficult than others to classify, that is, they receive substantially lower precision and/or recall for that category compared with the others. Reasons for this may be:

(1) very few positive training examples for the class, and/or

(2) lack of good predictive features for that class.

In the former case, there is little option but to obtain more training cases for that class. As a small consolation, classes that account for only a small fraction of the probability distribution can likewise have only a small effect on overall accuracy (or micro-averaged F-measure). However, just because a class has a small representation in the training set does not mean that in deployment the precision/recall for that class will not be important. For example, in a movie recommender system, although one may have only seen and ranked a few excellent movies and many mediocre ones, the precision for the minority class is most important.

Our hypothesis is that in case (2) above, where it is hard to get good predictive features for some class(es), existing feature scoring mechanisms will focus on the features that are useful predictors for other, easier classes, and will ignore the difficult classes—the 'siren pitfall.' This is exactly the wrong thing to do—difficult classes need, if anything, *more* attention and features selected for them, so that they can be better classified.

2.1 Empirical Demonstration of the Siren Pitfall

We further hypothesize that this pitfall occurs pervasively, even in well-balanced 'research quality' text classification problems. We demonstrate this by a detailed analysis on a balanced text classification task. As a running example, we take the task of classifying research paper abstracts into various fields of computer science. For the dataset, we extracted a set of abstracts and their associated classifications within the Cora topic categorization that was once made available by Whizbang.com. We selected from their many topics 36 classes, evenly distributed among 6 broad topic areas, in order that the task should be relatively uniform in difficulty (for comparison, an inhomogeneous task might have included a few classes from a branch of biology). To control for problem (1) above, we populated each class with exactly 50 training cases. The many classes make for a low majority-guessing accuracy, so performance above 1/36=2.8% accuracy is attributable to useful features, not chance. (The features are Boolean, indicating whether a specific word appears in the document, where a word consists of consecutive alphanumeric characters, forced lowercase, with no stemming and no stopword list. We include all words except very common words appearing in >25% of the documents, and rare words occurring in fewer than three documents.)

Even in a task with fairly homogeneous and uniform-sized classes, there is significant variation in the difficulty of the individual classes. To illustrate this, we measured the precision, recall, and F-measure for each individual class. (Precision P = true positives / all guessed positives. Recall R = true positives / all positives in ground truth. F-measure is their harmonic average = 2 P R/(P+R).) We used a state-of-the-art classifier: a multi-class one-vs-all Support Vector Machine (SVM, linear kernel, C=1) (Witten & Frank, 2000). We selected the top 500 features via IG (after determining 500 is sufficient for acceptable performance; see Figure 4). We used 4-fold stratified cross-validation to reduce random fluctuations in the measurements due to randomized splits. For each fold, the feature selector must be re-run from scratch so that no information leaks from the test set.

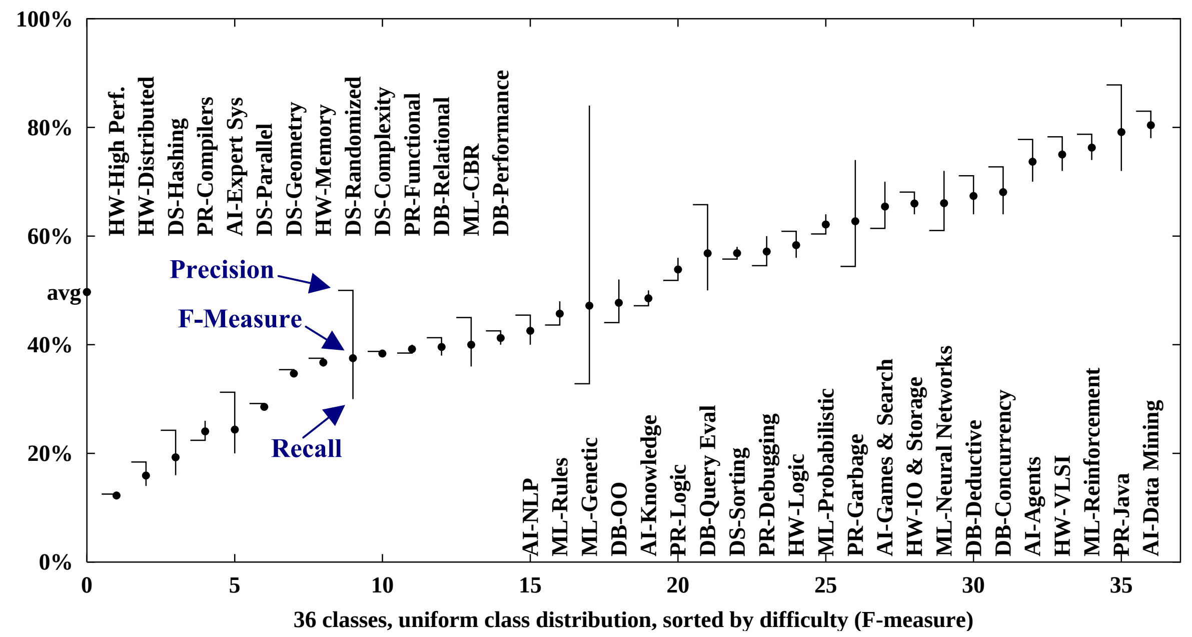


Figure 1. Baseline precision, recall and F-measure performance for each class of the Cora dataset. The 36 classes each have 50 training examples and are ordered here by their F-measure. The classifier—SVM on 500 features selected by Information Gain—obtained 50% accuracy (& F-measure) overall.

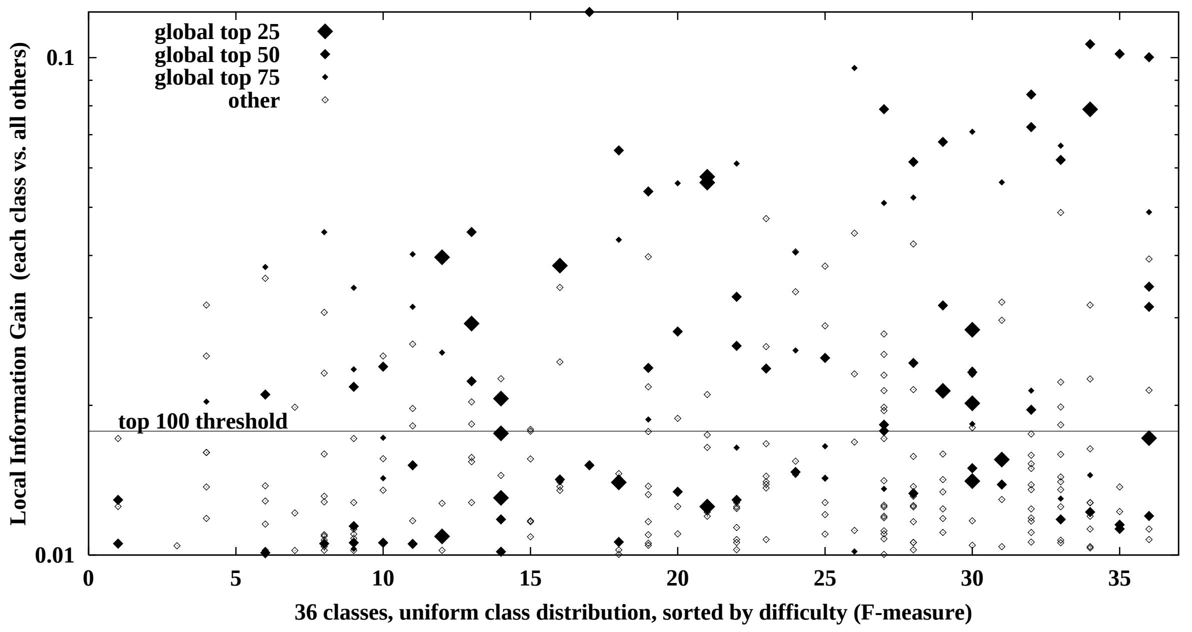


Figure 2. Information Gain scores of top features. In each column corresponding to the class ordering used in Figure 1, we plot the local Information Gain of each feature in distinguishing that class from all other classes. Additionally, we indicate the top *global* Information Gain scores via point shapes, e.g. large diamonds mark features included in the global top 25 features.

Figure 1 details the results. For each class, the whisker plot shows the precision (bent end), recall (straight end) and F-measure (round marker). The classes are sorted by their F-measure. Scanning the whiskers, we see that some classes have a wide discrepancy between their precision and recall.

The overall macro-average F-measure is 50%, indicated by the round marker on the y-axis. This is equivalent to the micro-averaged F-measure, because the class distribution is uniform. Micro-averaged F-measure is a per-document measure, so it is heavily influenced by larger classes, while macro-averaged F-measure gives equal weight to each class regardless of size. Since smaller classes tend to be harder to classify and there tend to be more of them than larger classes, the macro-averaged F-measure is often said to emphasize the smaller classes, while micro-averaged F-measure emphasizes the large classes. Research literature that studies only the micro-average, therefore, might more easily overlook this pitfall.

We see there is wide variation in the F-measures across the classes. In particular, notice the classes HW-High Performance and HW-Distributed have the worst scores. Since we have controlled for uneven class distribution (case 1), by our hypothesis, we expect that the available features for predicting these difficult classes are weak and excluded in feature selection. That is, they are weaker than the features available for other classes, and the feature scoring mechanism is drawn myopically to those stronger features.

To verify this, Figure 2 shows a plot of the IG scores of the top predictive features for each class independently. The scores for each class are plotted in separate columns, sorted left-to-right just as in Figure 1. The higher the dot, more useful the feature is to that class. Each column has all the features, though in different positions (some out of view below the x-axis). Notice in general that the more difficult classes toward the left tend to have fewer good predictive features.

In selecting features for the baseline in Figure 1, we computed IG on the global multi-class task, as typically practiced in the literature. To indicate the global selection order, we embolden points in Figure 2 that represent top features selected by the global feature scoring; a globally selected feature will embolden one dot in each column. We use different marker icons to label features that are included in the top 25, top 50, and top 75 globally selected features; features that were not selected within that number are represented by small dots. This gives a view into how the global feature selection mechanism is allocating features among the classes, and which locally predictive features it is omitting. Supposing independence of features for a moment, the more features selected higher up in each column, the more easily that class should be to discriminate by the induction algorithm. A column with none of its best features selected, e.g. columns 3 and 7, may then be more difficult to classify. For this reason, we truncate the graph below at 0.01, so our focus is only on *predictive* features.

Observe that there is wide variability between the classes in both the number of 'best' features, and the maximum or average scores of those features. None of the best features for column 7 have been selected globally. Columns 2 and 5 show no dots at all—their best features fall below the x-axis threshold chosen for this graph. These two classes also have some of the lowest F-measures. Finally, observe that the top 75 features do not appear to be 'fairly' distributed across the classes. This lends evidence to our hypothesis that the global feature allocation does not pay adequate attention to difficult classes.

Caveats: Looking at the allocation of features in this way does lend insight, but this evidence is not completely consistent. For example, class 15 shows a lack of predictive features, yet achieves a nearly median F-measure. Measuring IG locally is only a heuristic pointing towards the best features for each class, and may not be the optimal metric to use; resolving that question is outside our scope here, but see Forman (2003). Any 'imperfections' in the scoring metric slightly scramble both the order in which features are selected globally, and the local ranking of the feature dots in each column. Finally, features are not independent, and their value to the classifier depends on what other features are included.

2.2 The Siren Pitfall is Persistent—Non-Solutions

Instead of measuring the global IG, the 'obvious' solution to this problem is to measure the IG score for each class against the others as we have done, and then take the maximum score achieved by each feature for any individual class, a.k.a. Max.IG or Max.Chi (Yang et al., 1997). The horizontal line in Figure 2 represents the particular threshold to reap the top 100 features selected by this proposal. This would certainly do a better job of distributing the selected features among more classes, but note, however, that it still suffers from the very problem we are trying to solve. Observe that difficult classes, such as 1, 2, 3 and 5 have no features rated above this threshold, so they would still not receive any allocation.

Moreover, as a Gedanken experiment, should there be a class with many very strongly predictive features, it would serve as a lightning rod even for this 'max' variant of feature scoring, completely shielding the other classes from any allocation of features. For such a dataset, random feature selection may prove superior. Any random fluctuations or 'imperfections' in the scoring metric would help it overcome such a situation. This suggests that a random element in feature selection may prove beneficial.

Furthermore, this problem *persists* for any other feature selection metric and other aggregation functions, such as the mode, mean, clipped mean, or any general affine transformation.

Some have criticized the policy of selecting a fixed number of features for induction, and have instead proposed the selection method adjust the number of features chosen based on their distribution in the training data. For example, Rennie (2001), among others, propose using a significance threshold on the result of a statistical test. For such methods, one does not know *a priori* how many features will be selected for a new dataset, but for a given dataset, the significance threshold parameter effectively skims features off the top just as other scoring mechanisms, yielding no protection from the siren pitfall.

Although we have demonstrated the pitfall here only for Information Gain on a single text task, we have conducted similar experiments yielding similar observations with other feature selection metrics (including CHI, Max.IG, weighted Max.IG, and hypothesis testing) and with another text dataset (fbis of Han & Karypis, 2000). We omit their analyses for brevity.

3. A Family of Solutions

Desiderata for any solution to the siren pitfall include:

1. The presence of one or a few classes having many good predictive features should not hide the features useful for other classes. A (partially) randomized algorithm would satisfy this, e.g. randomized feature sets should be robust to the siren effect.
2. While the discussion above centered on a text task with a controlled uniform distribution, natural tasks tend to have significantly skewed class distributions. Large classes will affect the overall accuracy more than smaller classes. The algorithm should have the capability to allocate its attention appropriately. In contrast, small classes are often difficult to learn and may therefore have a greater need for more features.
3. It would be desirable if the solution were tunable when we have an estimate of the future testing class distribution, which may differ significantly from the training distribution.
4. Likewise, it would be useful if the solution were tunable for cost-sensitive classification.
5. It should be reasonably quick to compute for large numbers of classes and huge volumes of features. This rules out wrapper search methods.

3.1 SpreadFx[R,M] Feature Selection Family

The basic kernel of the solution applies round-robin turn taking to let each class propose features. To generalize a bit, we first propose an abstract family of feature selection algorithms that is parameterized across two general dimensions. Any family of algorithms raises more questions about what its optimal parameterizations may be. Nonetheless, they can be useful for dissecting a problem at an abstract level and considering options. Later we perform an empirical evaluation of the idea for three instantiations. We begin by presenting the family of algorithms:

```
Procedure SpreadFx[R,M](dataset) → ranked list of features
  for each class c of dataset:
    rank all features according to M for the binary sub-task
    of discriminating class c vs. all other classes combined.
    store this feature ranking for class c.
  while output not complete:
    call scheduler R to select a next class c.
    select the next feature fx from M's ranking for c.
    append fx to the output, if not already present.
```

The first parameter R, in its most general form, is a dynamic scheduling policy among the classes. We present here two of the simpler policies we studied:[1]

Round-Robin: This simple policy takes the best feature suggested from each class in turn.

Rand-Robin: This randomized scheduler, motivated by the observation on randomization above, selects the next class randomly with probabilities according to the class distribution. If the importance of the classes is unequal and known (e.g. classification costs, or more popular categories), this information could be used to skew the selection probability distribution.

The second parameter M is any feature-ranking method for two-class tasks. This could involve a feature scoring metric, such as Information Gain or Chi-Squared, but more generally, it need only return a total order of features for each class. Absolute scores for the features are not compared against one another, so different ranking algorithms might be used for different classes (supposing there were prior knowledge that particular classes would benefit from certain known methods, or we as a machine learning discipline eventually learn in which situations to apply various feature scoring metrics). Note that the feature ranking algorithm only need deal with a myopic two-class task, and so this opens up the possibility even on multi-class tasks to use methods that can only handle binary tasks, such as Odds Ratio (Mladenic et al., 1999) and Bi-Normal Separation (Forman, 2003). Note also that the sub-tasks may be computed independently and are amenable to parallelization. The results of the ranking are used in order, so online or anytime algorithms may be employed in addition to traditional batch algorithms. Finally, since the ranking is dependent only on the class and the dataset, it may be pre-computed and re-used, unlike algorithms that resample or permute the dataset and repeatedly call the ranking algorithm.

4. Evaluation

We begin by illustrating the improvement that SpreadFx[Round-Robin, IG] makes over traditional IG for the 36-class Cora dataset presented earlier. As before, we performed a 4-fold cross-validation on the dataset, using multi-class SVM and selecting the top 500 features. The arrows in Figure 3 show the gain in F-measure for each class. We see dramatic improvement for most classes, especially at the left, and a slight decrease for some of the easy classes on the right—a tradeoff we expect.

In order to obtain a single performance statistic for the entire dataset, we macro-average these individual F-measures. (This is equal to the micro-average F-measure

[1] Other variants we studied achieved unsurprising hybrid results and are omitted for clarity and brevity.

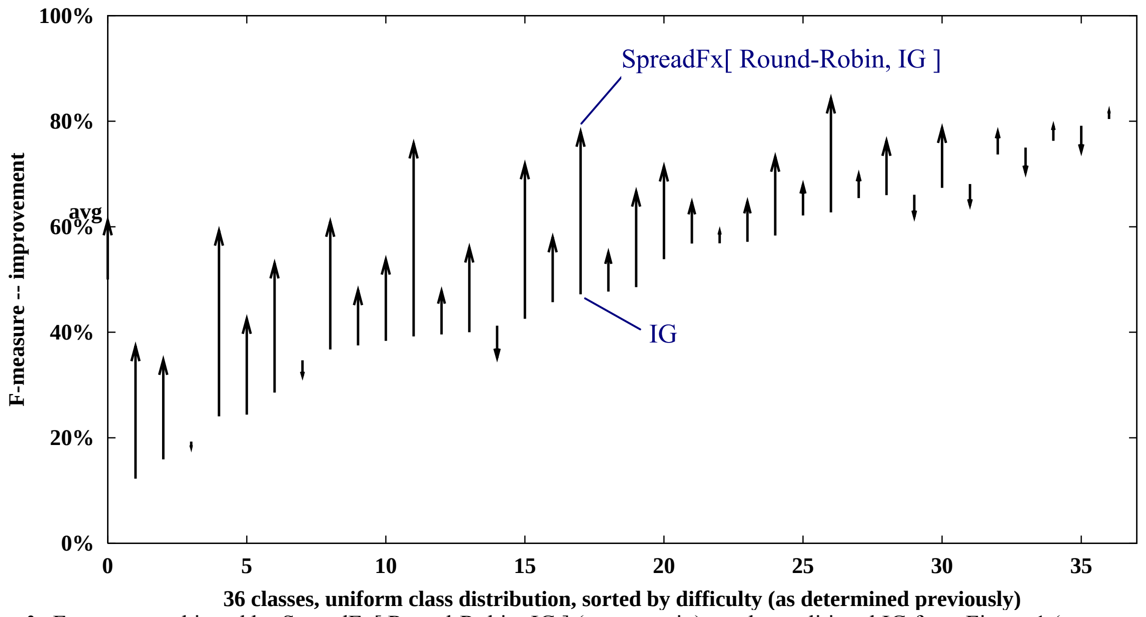


Figure 3. F-measure achieved by SpreadFx[Round-Robin, IG] (at arrow tip) vs. the traditional IG from Figure 1 (at arrow tail).

in this case because the class distribution is uniform.) In this experiment, we achieved an overall F-measure of 61.2%, up 22% from the previous baseline of 50% for traditional IG. (We repeated this experiment for Naïve Bayes and saw a similar improvement of 12% overall.)

Table 1. Benchmark Datasets

Dataset	Source	Docs	Words	Classes
Cora	Whizbang	1800	5171	36
fbis	TREC	2463	2000	17
La1	TREC	3204	31472	6
La2	TREC	3075	31472	6
Oh0	OHSUMED	1003	3182	10
Oh5	OHSUMED	918	3012	10
Oh10	OHSUMED	1050	3238	10
Oh15	OHSUMED	913	3100	10
ohscal	OHSUMED	11162	11465	10
Re0	Reuters	1504	2886	13
Re1	Reuters	1657	3758	25
tr11	TREC	414	6429	9
tr12	TREC	313	5804	8
tr21	TREC	336	7902	6
tr23	TREC	204	5832	6
tr31	TREC	927	10128	7
tr41	TREC	878	7454	10
tr45	TREC	690	8261	10
wap	WebACE	1560	8460	20

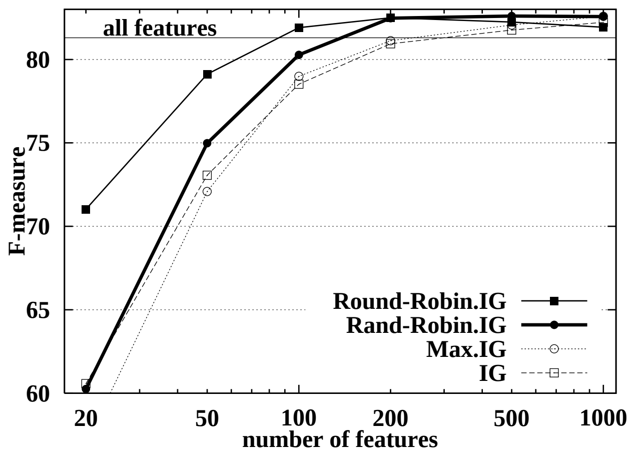
4.1 Improvement over all 19 Datasets

Next we present an evaluation over a large classification benchmark to test the merit of SpreadFx applied to the widely practiced IG and CHI methods. Certainly as we increase the number of features to a very large number, any feature selection algorithm will begin to provide many predictive features for all classes. So the primary hypothesis to test is whether the benefit of SpreadFx is beneficial at smaller numbers of selected features. That said, we would also like to see a gain for larger numbers of features selected.

For the induction algorithm, we chose the multi-class Support Vector Machine (SVM), as it is considered among the best in class for text classification, and quite popular (e.g. Yang & Liu, 1999; Joachims, 1998). We initially expected that it would be difficult to improve SVM results. To show that the results are not particular to SVM, we also demonstrate similar improvement for the traditional Naïve Bayes classifier, which is more highly sensitive to feature selection.

We performed our evaluations on the Cora dataset, plus 18 other text datasets provided by Han and Karypis (2000). Refer to Table 1. The classification tasks are drawn from standard benchmarks such as Reuters, OHSUMED, and TREC, among others. The datasets range from M=6 to 36 classes, 2,000 to 31,000 binary features, and have uneven class distributions with a median of 1 positive to 17 negative training examples (and average 1:31). No class is duplicated in different datasets. For a detailed exposition of the datasets, please refer to their paper or else Forman (2003). We will gladly make the feature vectors available on request.

For each dataset and feature selection scheme, we perform 4-fold cross-validation runs, obtaining the macro-averaged F-measure across all the classes of the dataset. We then average these results across five random stratified cross-validation splits for each of the 19 datasets. (The results for accuracy and even micro-



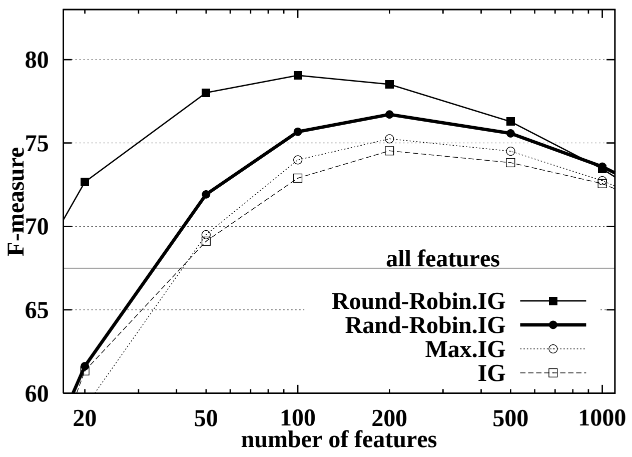
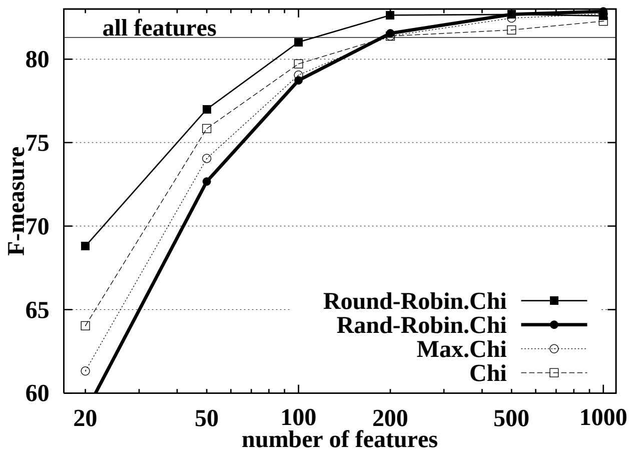


Figure 4. Macro-average F-measure for IG variants averaged over all 19 datasets for SVM (left) and Naïve Bayes (right).



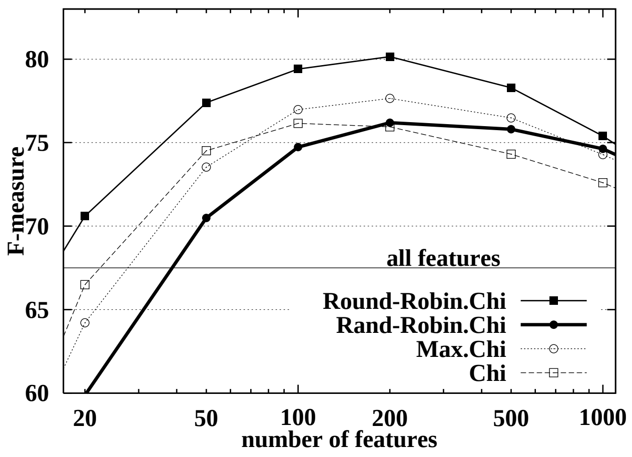


Figure 5. Same as Figure 4, but for Chi-Squared variants.

average F-measure are qualitatively similar to what follows and are omitted for brevity.)

Figure 4 presents the results for SVM (left) and the traditional Naïve Bayes classifier (right). Each graph shows results for the popular multi-class IG metric, as well as the Max.IG variant and SpreadFx variants for both of the scheduling algorithms discussed.

First, observe that traditional IG and Max.IG performed worse than either SpreadFx variant over most of the range for both classifiers. As we expect, the greatest gain appears at smaller numbers of features selected, with Round-Robin providing the best improvement. In contrast, Rand-Robin at 20 features is equivalent in performance to plain IG. The best performance over both graphs is obtained by SVM using Rand-Robin with 500-1000 features, whereas Round-Robin declines here. Observe that SVM with 100 features selected via SpreadFx[Round-Robin, IG] has better performance than IG's best performance with an order of magnitude more features, and is better than using all the features, indicated by the labeled horizontal line. That IG has difficulty achieving this level of performance may lend to the popular myth that support vector machines do not benefit from feature selection. Clearly Naïve Bayes is much more sensitive to feature selection, and Round-Robin leads to a great performance improvement.

To demonstrate that the siren effect is not peculiar to IG, we also present the results for CHI and its variants in Figure 5 likewise. The Round-Robin variant again dominates in protecting either classifier from the siren effect. Rand-Robin proved weaker paired with CHI.

5. Conclusion

For all multi-class feature selection methods that perform *independent* feature-scoring we have exposed a pitfall whereby they get distracted from selecting useful features for difficult classes if there is a supply of strongly predictive features for easier classes. We demonstrated this in detail on a dataset that has been carefully constructed to have a uniform class distribution and roughly uniform topical content in each class. Text classification tasks in real-world settings are rarely this regular, e.g. classifying email into folders, and would be even more likely to exhibit this siren pitfall.

We then discussed a parameterized family of algorithms to distribute the allocation of features among the classes, presenting two scheduling policies that are simple to implement. In evaluation on a substantial benchmark, we found consistent improvements for multi-class SVM and Naïve Bayes over basic IG or CHI, especially at smaller numbers of features selected. We note that SVM using features selected by SpreadFx[Rand-Robin, CHI] performed better with 500-1000 features than any other method, including using all the available features.

The proposed family of feature allocation policies attempts to be 'fair' in distributing attention to all classes, optionally according to their class distribution or other estimated cost weighting. We note that there is no guarantee that such a policy will work better. For example, suppose there were a large important class for which no features are predictive—a policy that focuses features on this large difficult class may ultimately suffer overall. It could be that allocating that budget of features to other classes would lead to a much greater overall improvement in classification. There are certainly few guarantees in this business. The best we can hope for is that typical text classification tasks rarely exhibit such pathological behavior, and so there may be some feature selection methods that are significantly better and more robust on typical text classification tasks encountered in practice.

Potential future work includes verifying the benefit for other promising classification models, other benchmarks including non-text and cost-sensitive scenarios, other scheduling policies, and other base feature selection metrics, such as weighted Bi-Normal Separation (Forman 2003), which can otherwise be applied only to two-class tasks. Finally, although we declared at the outset that wrapper methods are outside the scope of this paper, note that advances in fast scoring methods, such as proposed here, should be welcome to research in wrapper methods for use as potential heuristics to guide their search more efficiently.

Acknowledgments

We gratefully acknowledge the WEKA machine learning project for their open-source software (Witten & Frank, 2000), Han and Karypis (2000) for their prepared datasets, and Tom Fawcett for the Cora dataset. We acknowledge the INRIA Informatics and Distribution Laboratory for cycles on the ID/HP i-cluster. Thanks also to the reviewers who made this a better paper.

References

Forman, G. (2003). An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *Journal of Machine Learning Research*, *3*, 1289-1305.

Guyon, I., Weston, J., Barnhill, S., & Vapnik V. (2002). Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning, 46,* 389-422.

Hall, M., & Holmes, G. (2003). Benchmarking Attribute Selection Techniques for Discrete Class Data Mining. *IEEE Transactions on Knowledge and Data engineering*, *15,* 1437-1447.

Han, E.H.S., & Karypis, G. (2000). Centroid-Based Document Classification: Analysis & Experimental Results. *4th Conference on Principles of Data Mining and Knowledge Discovery* (pp. 424-431).

Joachims, T. (1998). Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *10th European Conference on Machine Learning* (pp. 137-142).

Kohavi, R., & John, G. H. (1997). Wrappers for Feature Subset Selection*. Artificial Intelligence, 97,* 273-324.

Mladenic, D., & Grobelnik, M. (1999). Feature Selection for Unbalanced Class Distribution and Naïve Bayes. *16th International Conference on Machine Learning* (pp. 258-267).

Mladenic, D., & Grobelnik, M. (1998). Word sequences as features in text learning. *17th Electrotechnical and Computer Science Conference*.

Rennie, J. (2001). Improving multi-class text classification with naive Bayes. *Master's thesis, Massachusetts Institute of Technology* (ch. 5).

Witten, I. H., & Frank, E. (2000). *Data mining: Practical machine learning tools with java implementations.* San Francisco: Morgan Kaufmann.

Yang, Y., & Liu, X. (1999). A Re-examination of Text Categorization Methods. *ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 42-49).

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