Machine Learning & Text-Based Classification

Heidy Khlaaf, Florida State University
Lucy Simko, Washington and Lee University
Emily Yu, Williams College

Introduction

Automatic text-based classification is a widely applicable and effective tactic used to quickly sort documents into a bevy of categories. Support Vector Machines (SVMs) have been empirically shown to be the most accurate when sorting documents between evenly-sized classes. However, they falter when asked to identify documents in a minority class because of the parity of minority training examples: it is often more costly to misclassify a minority instance than it is to misclassify a majority class instance when the goal is to identify instances of the minority class. We test both known and novel methods of feature selection and propose using a hierarchy of SVM classifiers to more accurately select members of the minority class. Our conjecture is that transformations of the problem which make the resulting feature sets more well-defined will reliably improve accuracy.

We used the uneven distribution between male and female computer scientist to typify the minority class problem; our objective was to identify the homepage of every female computer scientist. We first built a customized web crawler which retrieved pages linked at maximum depth n (we used $n = 4$) from the computer science web-page of any given university. Using loose heuristics, it attempted to eliminate non-faculty web-pages, but only approximately half of the pages returned belonged to computer science faculty members. For details, see the section on the customized crawler.

The problem of identifying web-pages belonging to female computer scientists presents several unique issues. Because of the skewed distribution of data, we had to be careful that the classifier was not simply identifying all the instances as the minority class and therefore achieving a fairly high accuracy.

The nature of the data—web-pages, not data with a specific jargon—meant that the feature sets on which the SVM(s) was trained were too massive to be practical, so we had to employ effective feature selection. We concentrated on an established heuristic in information retrieval, *TD-IDF*, and propose a new method, *use-ratio*. *Use-ratio* is calculated based on class occurrence, whereas *TF-IDF* does not take into account the frequency of a term between classes. For details on the calculation of these numbers, see *Approach*.

We suspect that *TD-IDF* is extremely useful in tasks involving a specific query precisely because the machine relies on the terms provided in the query to classify documents as 'relevant' or 'non-relevant,' not on information learned from a training set—so it is intuitively correct to employ a heuristic that eliminates terms based on their frequencies over *all* documents, not just over either 'non-relevant' or 'relevant' documents. In text-classification problems, however, the situation is very different, because the machine has learned from a training set the characteristics of the required classes and is not typically given a query or set of important terms. Our heuristic, *use-ratio*, attempts to identify the terms most specific to each class and eliminates all others from the document vectors used in training and classification. Our experiments indicate it is not clear that *use-ratio* is a better choice than *TD-IDF* in all situations, but future work could investigate both the effect of feature set size in relation to selection heuristic and the effectiveness of selection heuristics on data sets with different characteristics (*i.e.* data sets with specific jargon, extremely small data sets, *etc.*).

As another strategy, we considered how well the feature sets reflected the two classes and determined that a hierarchical system of SVMs should drastically improve accuracy. A key feature of the hierarchy is not that the two classes at any given level should be equal in size, but that their feature sets should be sufficiently distinct. In contrast to changing the selection technique, a two-classifier hierarchy with the levels *cs-faculty vs. non-cs-faculty* and then (of the documents classified as *cs-faculty*), *male vs. female* shows potential because the *male* vs. *female* averaged around 91% accuracy overall.

The differences between the characteristics of the classes are significant: there is less distinction between *cs-faculty* and *non-cs-faculty* than between *male* and *female*, because although *male*, *female,* and *cs-faculty* are categories, *non-cs-faculty* is the absence of a category and therefore has very few (if any) distinguishing features. *Use-ratio* performed better than *TD-IDF* in the first level of the hierarchy, in which the categories had few distinctive features, and equally with *TD-IDF* on the second level of the hierarchy. This lead us to surmise that the way *use-ratio* selects features can help compensate for lack of distinction between the feature sets of two classes and, although it performed accurately with our well-defined classes, it does not improve accuracy over the *TD-IDF* heuristic.

Approach

A standard approach in binary text classification is to represent each document using the bag-of-words approach and then apply an

SVM with a linear kernel.[1] Recall that our task is to discriminate female computer scientists' web-pages from all the other pages returned by our customized crawler. Because the crawler is not completely accurate, the ratio of female computer scientists' web-pages to others' is one to fifteen.

In text classification, one typically preprocesses data by stemming the feature set, removing stop words, and removing words with a low *TF-IDF* score [term-frequency * inverse-document-frequency].[2] In theory, eliminating words with low *TF-IDF* scores removes the features that are least helpful to the SVM, as defined by other information retrieval problems.

Because our classification problem is contingent upon the classifier's ability to distinguish the gender of the subject or author of a web-page, we assume that there is a significant difference in the vocabulary of women computer scientists. We anticipated that the classifier would rely upon obvious gender-specific words like 'he', 'she', 'him', 'her', and less obvious words that are also used unequally between the genders. Nevertheless, all gender-related words had to be preserved, and this necessitates a customized stemmer and stop word list. We implemented a modified version of the Porter stemmer[3] and further changed it to stem pronouns to their subject forms and deal with other varied abbreviations that frequently appear in faculty homepages, such as ‘grad.’ to ‘graduat’ or 'assoc.' to 'associate'. Accordingly, we customized the stop word list, adding to the generic list of pronouns and conjunctions words commonly found on gender- and field-ambiguous homepages, like “pdf”, “et”, “ad”, “me”, “you”, et cetera, and removing any form of the third person singular pronoun.

To ameliorate the skewed distribution, we replicated the female computer scientists' web-pages by fifteen times. An initial test in which we set the TF-IDF threshold to 1.5[4] resulted in an accuracy of 57.8%. Our conjecture is that the low accuracy is caused by both the lack of definition for the majority class and the skewed distribution. The majority class—any web-page not belonging to a female computer scientist—is not intuitively distinguished by a group of words, whereas the minority class seems more likely to contain highly definitive features. Although replication helps increase minority class accuracy, it cannot substitute for a more diverse concentration of minority class

---

[1] T. Joachims, *Transductive Inference for Text Classification using Support Vector Machines.* Proceedings of the International Conference on Machine Learning (ICML), 1999.

[2] Joachims, 1999.

[3] http://snowball.tartarus.org/algorithms/english/stemmer.html

[4] Our decision to set the initial *TF-IDF* threshold at 1.5 was motivated by the size of our data set and the amount of our computing power. Because our data set was comprised of about 6,000 documents, the largest features set we could realistically give the SVM was about 5,000.

instances. Replication cannot change the SVM's margin more accurately than new instances or less skewed data can because the replication technique clones errors or outliers as well as normal instances.

To increase performance, we divide the classification problem into a hierarchy consisting of two steps: a basic problem of whether a web-page belongs to a computer scientist, and a second step to determine whether the computer scientist is male or female. Using a hierarchy of SVMs is a common solution when there is a pre-existing hierarchy among the categories about which the user cares [Akbani et al. 2004; Granitzer 2003; Sun and Lim 200; Dumais 2000; Liu et al. 2005][5]. Our dataset does have a natural hierarchy, albeit a small one.

Feature Selection

The average total number of unique stemmed words in the gender training data is 20,177; in the faculty training sets average 49,249 stemmed words. Running an SVM on such a large dataset was a challenge in our computing environment; thus, we investigated automated feature selection methods. We propose a new metric, *use-ratio*, which favors features whose use is concentrated in one of the categories, as opposed to the *TF-IDF* measure, which selects features concentrated in one document. Our conjecture was that although the *TD-IDF* measure works well for retrieving individual documents, *use-ratio* would select features more helpful to the classifier because it is calculated based on the frequency of the feature between the categories, not the document set. It is given by the equation $R = |Tc_1 - Tc_2| / T$, where R is the *use-ratio*, Tc_n is the total number of occurrences of a word in class n, and T is the total number of times a word occurs in all of the data sets.

One of the primary differences between the data in the two classification problems is that the gender web-pages are intuitively divided by a set of words, some of which are obvious, because the vocabulary is specifically tied to the gender of the subject of the web-page—that is, the feature set is by definition distinct for each class. In this case, the classifier must choose between two well-defined categories, as opposed to one well-defined category and whatever is not in that category. This property makes the gender classification problem a good candidate for our approach.

In addition to restricting the feature set based on the *use-ratio*, we eliminated words only used once. Because these words by definition had a *use-ratio* of 100%, this effectively reduced the number of features and eliminated the less helpful ones by acting as a tie-breaker. The elimination of features based on total count is a

[5] See References

successful heuristic used in other classification problems (see works cited). To compare the effectiveness of *use-ratio* and *TF-IDF* as feature selection heuristics, we created equally sized feature sets using these methods.

Customized Crawler

Our web crawler is built using an open source HTML parser[6] which fetches pages, parses the HTML and extracts links.[7] Because of the domain, we modified the parser to have a depth parameter *n*, which indicates that it crawl only *n* pages from the computer science homepage. Empirically, we observed that faculty and graduate student pages are likely to be within four mouse clicks away from the homepage, thus we sent *n* to be four. A depth of $n=4$ eliminated approximately 50% of false positives.

We modified our crawler to use heuristics for finding computer scientists' homepages. After extensive study of computer scientists' pages from several colleges and universities, a set of keywords in specific locations on a page that seemed indicative of being on a computer scientist's homepage or biography emerged. The main areas of interest on a web page are the headers (denoted h1-h6) and title because they provide the most information about the general content of the page. Finding words like "homepage", "biography", "professor", and "profile" in these areas show to be a decently accurate predictor that the page does indeed belong to a faculty member or student.

A second heuristic focuses on the content of the links. In some cases, the link text or URL indicates that the link leads to a computer science faculty homepage. For example, if the link text or URL contains some form of the word "biography" (i.e., "bio", "biographical"), the page the link points to is likely to be the biography section of a homepage. In other cases, the link text or URL do not indicate that the link is a homepage, but rather that the page containing the link is a homepage. For example, a page with a link to a resume or curriculum vitae (CV) is likely to be a homepage while the page the link leads to is not.

While these heuristics allow the crawler to find most faculty and graduate student pages, approximately 50% of pages identified as homepages are false positives. Around 20% of these false positives can be filtered out by making a list of words (e.g., "sitemap", "publications") that, if found in the URL, clearly indicate that the page is not a homepage or biography, but due to the fact that a formal set

[6] http://code.google.com/p/crawler4j/

[7] Our first attempt used an open source, generic web crawler called Crawler4j, but this was not sufficient because 1) it was not expressive enough to incorporate our specific criteria for selecting useful pages and 2) could not be halted prior to the point where it ran out of links to visit.

of rules for HTML formatting does not exist, as well as the fact that there is no standard for what a computer scientist's home page or biography looks like, it is impossible to create a perfect set of heuristics to find only pages belonging to faculty members or students, much less only the pages belonging to female computer scientists. Inevitably the crawler's output will contain false positives, thus our goal became to minimize false negatives, knowing that this would cause an increase in false positives. The specificity of the problem makes it highly unlikely that we can rely solely on a web crawler to find all the pages belonging to computer scientists and only the pages belonging to computer scientists. We therefore turn to machine learning techniques to attempt to solve the problem.

Experiments and Results

To test which feature selection method performed best on our data and to evaluate whether we should use a hierarchical approach we performed a four-fold cross validation over the labeled data. In Table 1 and Table 2 we show the distribution of the data.

Table 1: Faculty versus Non-Faculty instances

Instances	*First- Fold*	*Second-Fold*	*Third-Fold*	*Fourth-Fold*
Training	3895	3897	3895	3895
Testing	1299	1297	1299	1299

Table 2: Female versus Males instances

Instances	*First- Fold*	*Second-Fold*	*Third-Fold*	*Fourth-Fold*
Training	1272	1249	1290	1241
Testing	412	435	394	443

For each experiment we first ranked the features with respect to the feature selection metric (e.g., *TD-IDF*) and then chose the top ranked 500 features to use in an SVM with a linear kernel. We used grid search to optimize the parameters of the SVM over the training data in each fold of the CV. In the following tables we report the mean accuracy and standard deviation across the four folds.

Table 3 shows the results of a non-hierarchical approach in which we treat the task as a binary classification problem in which one class is all web-pages classified as Female CS faculty and “other”. Thus “other” contains Male CS and non-people pages. All three feature selection methods performed only slightly better than random guessing. However if we look at the accuracy by class we see that it is far more accurate at identifying female faculty than “other”.

Table 4 shows the result of the first step in a hierarchal approach: classifying each web page as faculty versus non-faculty. For this dataset, we see that the *Use-Ratio* feature selection method performs best. The second step of the evaluation of the hierarchical approach is to take all of the data correctly classified as faculty and retaining the same four folds, to now learn to discriminate male versus female faculty. The *Use-Ratio* feature metric for our Female classifier was sorted to minimize the feature set to 5,000 words .These results are shown in Table 5.

Table 3: Female versus Other classifier

Feature Vector	*Average Accuracy*	*Standard Deviation*	*Female classified*	*Other Classified*
Count	57.47%	1.26	294 out of 312	2012 out of 3584
IDF	57.87%	1.78	307 out of 312	559 out of 3584
Use-Ratio	56.28%	2.13	294 out of 312	1810 out of 3584

Table 4: Faculty versus Non-Faculty

Feature Vector	*Average Accuracy*	*Standard Deviation*	*Faculty Classified*	*Non-Faculty Classified*
Count	53.69%	2.11	173 out of 2384	2616 out of 2803
IDF	53.72%	1.47	31 out of 2384	2759 out of 2803
Use-Ratio	66.96%	2.79	1684 out 2384	1794 out of 2803

Table 5: Female versus Male Faculty

Feature Vector	*Average Accuracy*	*Standard Deviation*	*Females Classified*	*Males Classified*
Use-Ratio	90.1%	1.39	171 out of 303	1346 out of 1381

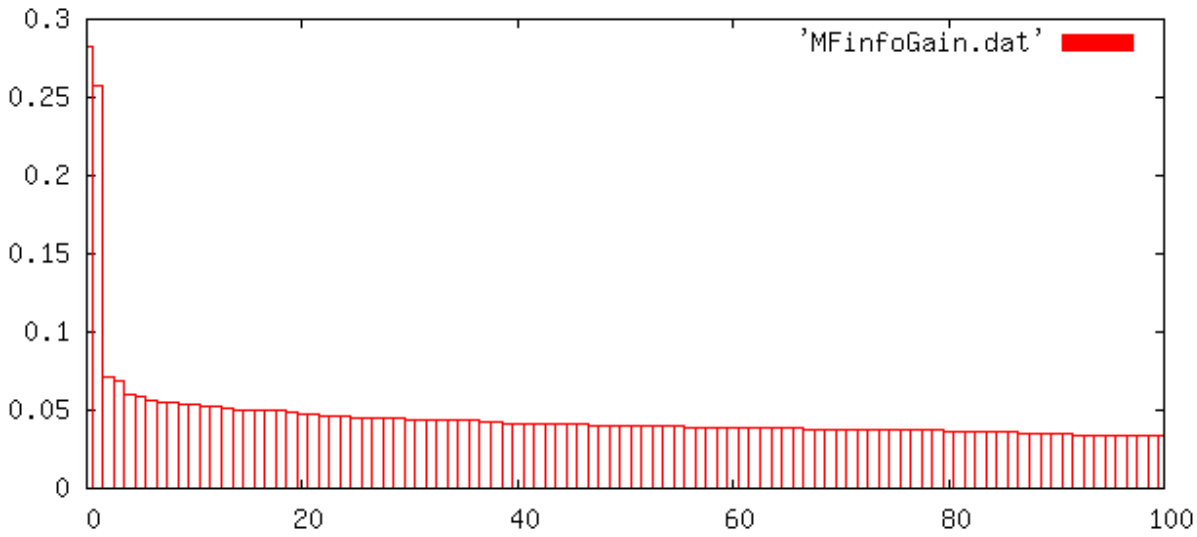
We experimented with several different feature selection strategies in order to optimize the classification accuracy. In our experiments, we assessed the entirety of our gender data which consisted of 2,385 instances. All feature metrics were sorted to minimize the feature set to 5,000 words; without using a feature selection strategy our feature set was compromised of 20,000 words.

Table 6: Females versus Male Faculty

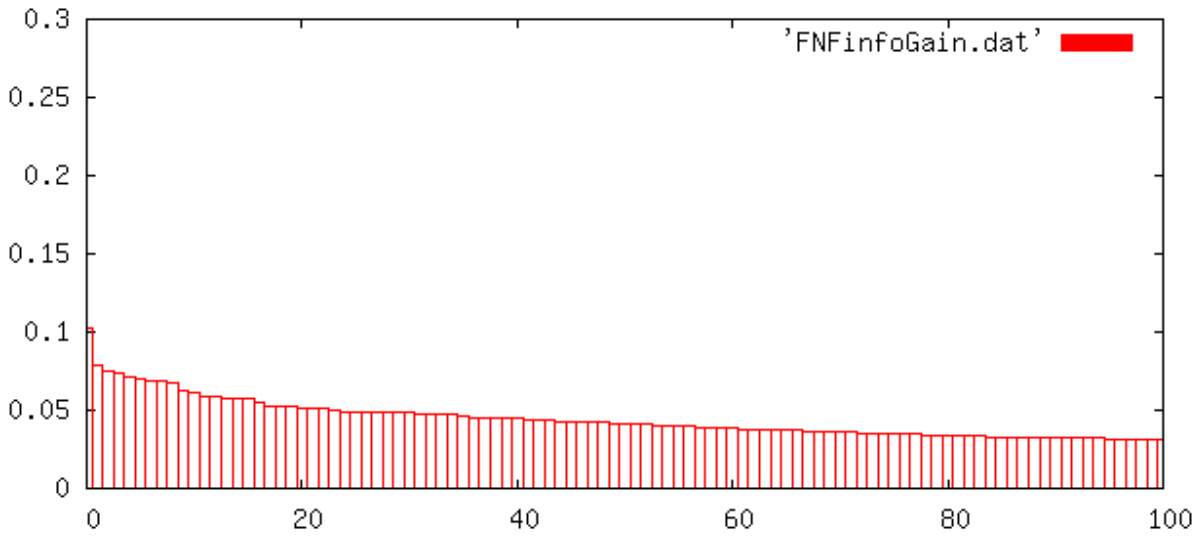
Feature Vector	*Average Accuracy*	*Standard Deviation*	*Faculty Classified*	*Non-Faculty Classified*
Count	90.13%	1.00	272 out of 411	1878 out of 1974
IDF	92.70%	1.58	279 out of 411	1920 out of 1974
Use-Ratio	92.18%	0.76	289 out 411	1910 out of 1974
None	90.96%	1.14	253 out of 411	1917 out of 1974

As shown in Table 6, *Use-Ratio* and *TD-IDF* perform slightly better than using all of the features (“None”), or choosing the most prevalent features (“Count”). We hypothesized that the significantly higher accuracy of the female vs. male classifier relative to the faculty vs. non-faculty is due to the well-defined keywords contained within the female class. This led us to utilize the Information Gain equation to calculate the weights of the top 100 features for the female and Faculty classifier. We used the *Use-Ratio* feature set for both our faculty and female feature set seeing that it provided the highest results for our Faculty classifier. The first graph below demonstrates that there exist two features, “he” and “she” that are weighed heavily. The second graph shows that there are no well-defined weights when determining the Information Gain for our Faculty classifier.

Weights of feature vector using Information Gain on Computer Science Female classifier

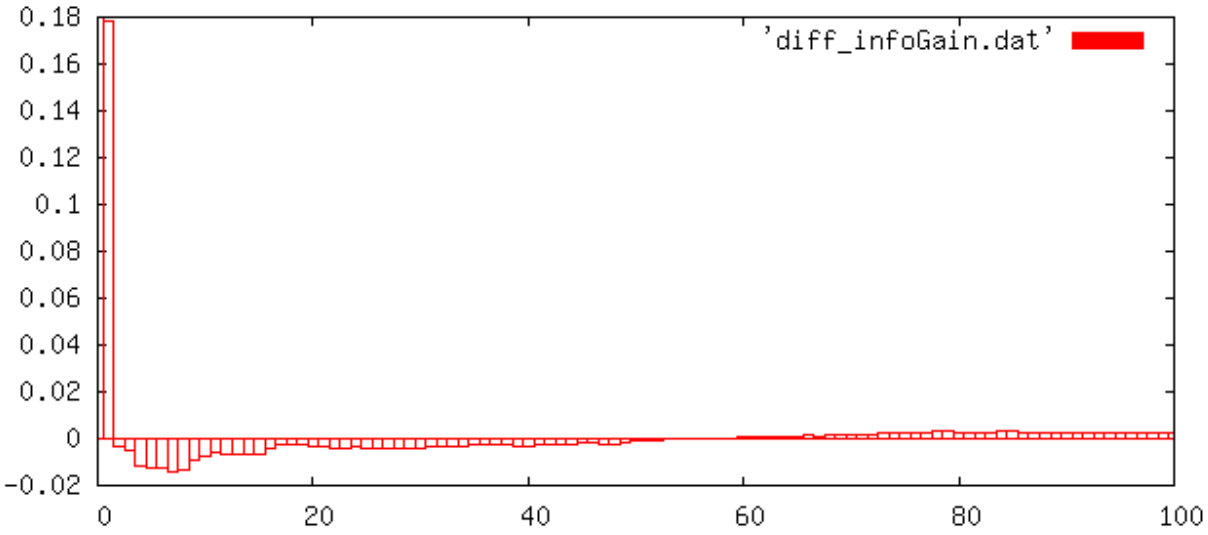


Weights of feature vector using Information Gain on Faculty classifier



The well-defined keywords found within the female feature set accounts for the high performance of our SVM classifier, while the faculty feature set demonstrates no well-defined features that would assist our classifier in properly differentiating between faculty and non-faculty. The graph below exhibits the differentiating results of our faculty and female feature sets which provides a clear mapping of the distinct feature sets, thus classification, of our Female classifier.

Difference of weights of Female classifier versus Faculty classifier



Conclusion

There exists an evident distinction between our faculty class and

our gender class. Table 3 demonstrates that an attempt to merge both classes into one classification problem would at best produce low accuracy rates. Comparatively, our Gender classifier achieved soaring results despite the lacking accuracy of the Faculty classifier. In our experiments pertaining to the split hierarchy, we used various feature metrics in order to achieve the highest accuracy possible. In the first level of hierarchy, *Use-Ratio* showed a significantly higher performance than that of the *TD-IDF*'s, while in the second level of our hierarchy, *TD-IDF* showed an equivalent performance relative to its *Use-Ratio* counterpart. The fact that our top hierarchy had a lack of distinctive features relative to the second, demonstrates that the *Use-Ratio* metric could compensate for a lack of distinction contained within a feature set. This does not, however, indicate that *Use-Ratio* performs more accurately with well-defined classes; our results have shown that there exists no improvement when compared to the *TD-IDF* heuristic. Other primitive methodologies such as *Count,* have shown no significant results or improvements on the feature sets.

As shown by our results using the Information Gain equation, a class that embodies a well-defined feature set exceeds in performance. The high accuracies of our Female classifier provide an incentive to improve the performance of the first level of hierarchy thus an improvement to the entire hierarchy. Although the lack of well-defined keywords within the faculty feature set could account for its low performance, various factors could be liable as well. We have neglected to investigate both the effect of feature set size in relation to selection heuristic and the effectiveness of selection heuristics on data sets with different characteristics (i.e. data sets with specific jargon, extremely small data sets, etc.). Future work pertaining to the faculty classification problem would incorporate these various factors.

References

Akbani, R., Kwek, S., & Japkowicz, N. (2004), "Applying support vector machines to imbalanced datasets*," Proceedings of the 2004 European Conference on Machine Learning (ECML) (pp. 39-50)*. LNAI 3201

Granitzer, Michael, "Hierarchical Text Classification using Methods from Machine Learning" (Masters Thesis, Institute of Theoretical Computer Science, Graz University of Technology, A-8010, Graz, Austria, 2003)

Dumais, Susan, and H. Chen, "Hierarchical Classification of Web Content," *Proceedings of SIGIR'00*, (2000), 256-263.

Joachims, T, "Transductive Inference for Text Classification using

Support Vector Machines," *Proceedings of the International Conference on Machine Learning (ICML)*, 1999.

Liu, Tie-Yan, et al., "Support Vector Machines Classification with a Very Large-scale Taxonomy," *SIGKDD Explorations*, *Volume 7, Issue 1* (2005), 36-43.

Sun, Aixin, and Ee-Peng Lim, "Hierarchical Text Classification and Evaluation," *Proceedings of the 2001 IEEE International Conference on Data Mining,* (2001), 521-528.

Yasser Ganjisaffar. Crawler4j: An Open Source Web Crawler for Java [http://code.google.com/p/crawler4j/]. Irvine, CA: University of California, Irvine, School of Information and Computer Sciences.