ASCERTAINING THE RELEVANCE MODEL OF A WEB SEARCH-ENGINE

BIPIN SURESH

Abstract

We analyze the factors contributing to the relevance of a web-page as computed by popular industry web search-engines. We also attempt to discover the underlying ranking model used by such search-engines by fitting known positive and derived negative examples for a set of queries.

Introduction

As a marketing strategy for a web-site's success, web-site designers have discovered that ranking highly in popular web search-engines like Google[1], Yahoo![2] and MSN Live[3] has a large impact on revenue.

Web search-engines adopt a variety of features (like the page's link-structure, description, site-hierarchy, freshness and PageRank™) to rank pages for a given query. The score from the ranking function that a search-engine uses is hoped to be an approximation of the true relevance of the page. Google claims it uses about 200 such features[4] to come up with a final score. The entire set of features and their relative weights, however, are left undisclosed.

Consequently, the problem of what features to tune a page on to ensure it ranks well in these search-engines has turned into one of half-guessing by industry experts[5] and adherence to generic guidelines[6] published by the search-engines themselves. Search-engine optimization (SEO) has thus become a lucrative venture, estimated to be $643 million in 2006[7].

In this paper, we aim to determine the relative weights of these features by using a machine-learned classifier. Our approach will be to model the ranking of a single commercial web search-engine.

In Section I, we do preliminary analysis of the data that we have procured. In Section II, we train a decision tree based model to come up with a discriminator between pages that are ranked highly and those that are not, purely on the features that we have chosen. In Section III, we try to make the decision boundary tighter by contrasting the results of two popular search-engines, and interpret the rules learnt in Section V.

Section I: Dataset Generation and Analysis

Dataset generation for the paper was obtained in two parts:

In the first part, we obtained pages that a current industry web search-engine considers to be of good quality for a certain set of queries. We did this by obtaining the top-100 web-queries of the year 2006 from Yahoo! Buzz[8], and querying the search-engine for the top 10 results for each query. This gave us 1,000 pages which we shall henceforth consider to belong to the positive class.

In the second part, we extracted 1,000 random web-pages from the open web-directory DMOZ[9], after making sure that none of them belonged to the first class. We shall henceforth consider these to belong to the negative class.

We then built a web-crawler to crawl and extract information about the web-pages. We also used the Yahoo! Site Explorer[10] which provides a means to collect information about the link-structure of a page, including the anchor-text associated with each page (which is the concatenation of the anchor-texts of all links pointing to the page in question).

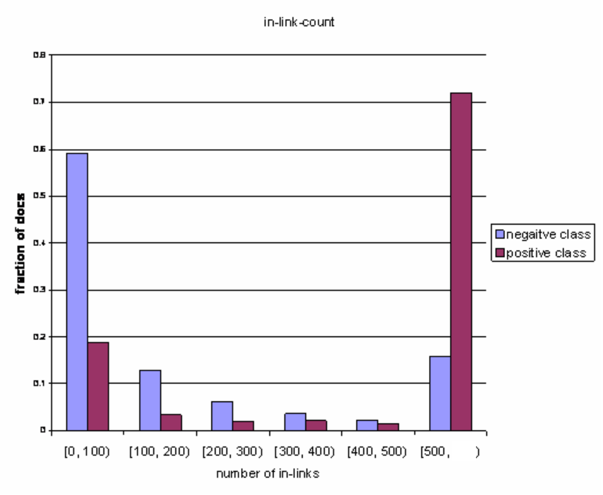

Figure 1: in-links
Plot indicates that positive-class documents are connected to by a large number of other documents.

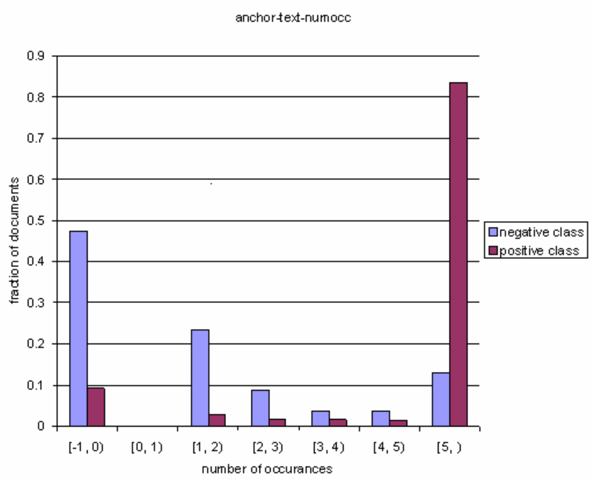

Figure 2: number of matches in anchor-text
Plot indicates that a large number of documents refer to positive class documents with the query-term as the anchor-text.

In this paper, we attempt to completely characterize the relevance of a document as a function of its primary attributes. Primary attributes are defined to be those which a web-site creator had control over – entities such as the link-structure and the content of the web-page. We do not consider derived attributes like PageRank™ (it is interesting to note that there are conflicting views of whether there is any discernable correlation between the link-structure of a page and its PageRank™[10][11]). There are two motivations for this choice – one being the hope that we can describe a process for web-page creators to improve their rankings without having to resort to optimizing third-party functions and; two, the fact that most of these metrics are usually not made public by the commercial web search-engines in any case.

Before building a model out of the data-set, we did some preliminary analysis to understand the nature of the distribution of our features across the two classes.

We first define an out-link to be any hyper-link from the web-page of interest to another URI on the web. We also define an in-link to be a hyperlink from any URI on the web to the web-page. In our current implementation, we do not differentiate between the source (or destination) of the out-links (or in-links) – internal links (links within the same website) are treated just as equally as external links (links to external websites). Figure (1) plots the distribution of the number of in-links. The distribution seems to suggest immediately the well-ranked pages seemed to be better connected and recognized by other web-documents.

For the next set of features, we first determined the following for every document that matched a query: the position of the first occurrence of the query in the title and text of web-page; the number of occurrences of the query in the title and text of the web-page and; the number of occurrences of the query in the anchor-text associated with the web-page. (A value of '-1' was assigned in case there was no match.)

We have plotted the distribution of these features across the two classes in Figure (2), (3) and (4). The number of occurrences of the query within the anchor-text already seems to show a strong relationship with the class of the document. The correlation is however not linear – Pearson's correlation coefficient for the data was a low 0.24. We will explore whether the relationship will be caught by our Machine Learned model which we shall derive in the next section.

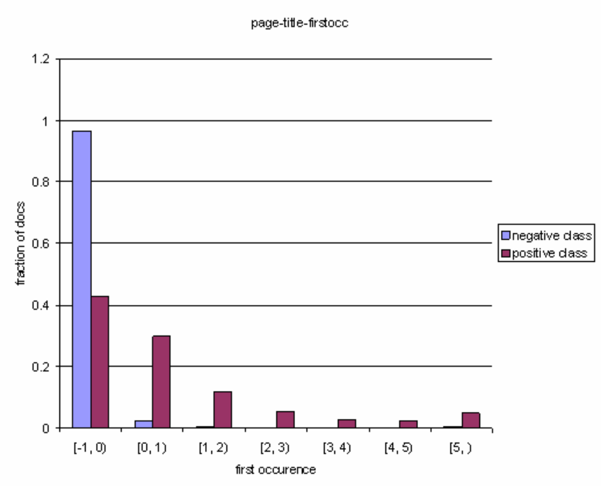

Figure-3: First occurrence of the match in the page-title
Plot indicates that negative class documents usually don't match in the title of the page.

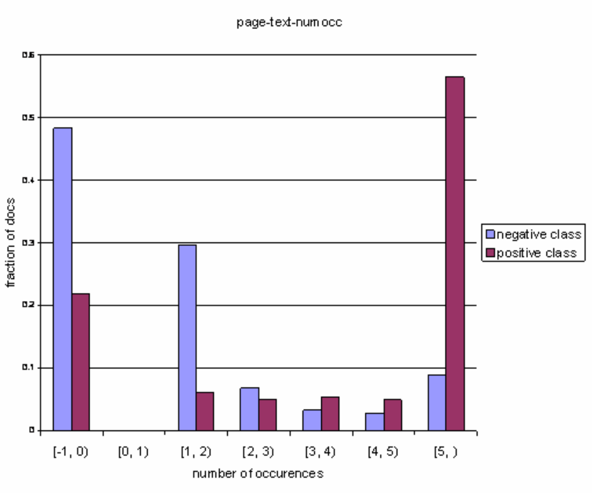

Figure-4: Number of occurrences of the match in the page-text
Plot indicates that positive class documents usually have quite a few matches in the body of document.

Section II: Classification

To discover an optimal discriminant rule for our dataset, we chose to train a Decision Tree based classifier model. This model was chosen based on two reasons: because of literature hinting that web-search engines actually do use some form of decision-trees to produce the final ranking functions[17] and; because of its learning can be easily interpreted as human-readable rules.

In the first experiment, we considered the following features:
(1) anchor-text-match: binary valued feature which indicates whether or not the query term is present in the anchor-text of the document.
(*2) page-title-match*: binary valued feature which indicates whether or not the query term is present within the 'title' tag of the page's HTML code.
(*3) page-text-match*: binary valued feature which indicates whether or not the query term is present in the document-text.
(*4) in-link-count*: number of documents linking to this web-page.
(*5) out-link-count*: number of documents that this web-page links to.

Since the training sample was relatively small (1000 positive class examples, and 1000 negative class examples), we chose to run a k-fold cross-validation process, with k=10. Table (1) displays the recall-precision matrix from one of the folds.

The classifier performs reasonably, with a macro-averaged recall of 84.62% and a macro-averaged precision of 84.06%.
Figure-5 shows the final rule learnt by the classifier. Upon inspection, it is noticed that the classifier prefers query dependent features (matches in various fields) to query-independent features (like the connectedness of the site).

Considering how important the query-dependent features seemed, we decided to explore a few more of them. We introduced 5 more features to the set that we already had:
(1) *page-title-firstocc*: the position of the first match of the query-term in the title of the document. It was a 0 based system, where a value of '0' meant that the query-term began at the very first position. A value of '-1' was assigned if there was no match in the page-title.
(*2) page-title-numocc*: the number of times the query-term matched in the title of document. A value of '-1' was assigned if there was no match in the page-title.

	Precision	Recall
Positive-class	89.5%	81.7%
Negative-class	78.7%	87.5%

Table-1: Precision and recall for classifier based on link and match features.

	Precision	Recall
Positive-class	95.8%	88.5%
Negative-class	86.4%	95.0%

Table-2: Precision and recall for classifier based on link and occurrence features.

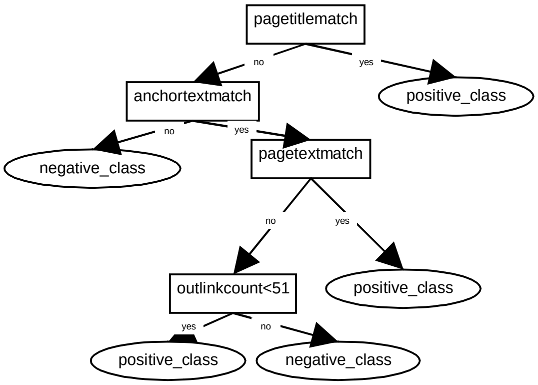

Figure-5: Decision tree with link and match features.

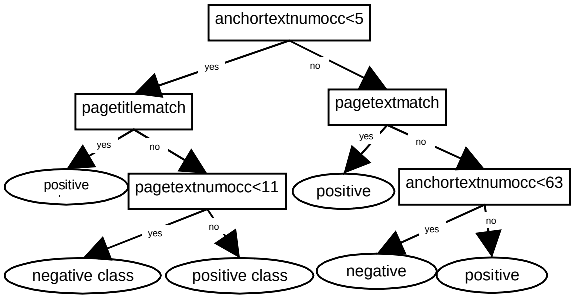

Figure-6: Decision tree with link and occurrence features.

(3) *page-text-firstocc*: same as *page-title-firstocc* feature, except that this time the matches were examined in the page-text.
(4) *page-text-numocc:* same as *page-title-numocc* feature, except that this time the matches were examined in the page-text.
(5) *anchor-text-numocc*: same as *page-title-numocc* feature, except that this time the matches were examined in the anchor-text.

We did not consider *anchor-text-firstocc* since the anchor text field was just a concatenation of all the anchor-texts of documents pointing to the page in question, and thus the first occurrence of a match in that field wouldn't contain any useful information.

Armed with these five new features, and the five from the previous experiment, we retrained the decision tree.

The overall performance of the tree improved with these new features, as shown by the recall-precision matrix in Table (2). The macro-averaged precision and recall numbers improved to 91.10% and 91.73% respectively.
The learnt rule was also updated, as depicted by Figure 6. However, the general trend of the classifier preferring query-dependent features over query-independent features still remained.

Section III: Making the decision boundary tighter

Having attained reasonable accuracies in the classification of well-ranked pages of a search engine versus randomly chosen pages, we attempted to make the decision boundary tighter. We did this by modifying the data-set – the positive class was, as before, the top-10 results for a given set of queries of a commercial web search-engine. The negative class however was changed to be the top-10 results for the same set of queries of *another* web search-engine which did not feature in the top-20 results of the first web search-engine. Thus, the negative class had results which the first web-search engine thought weren't good enough, but the second did.

	Precision	Recall
Positive-class	75.9%	94.2%
Negative-class	91.9%	68.9%

Table-3: Precision and recall for Google vs. Yahoo! Search classifierer.

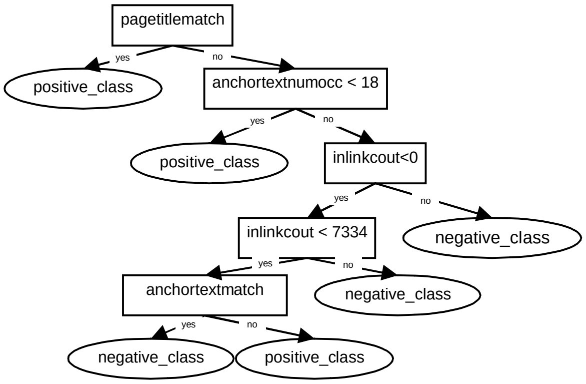

Figure-7: Decision tree discriminating Google and Yahoo! Search results..

In our experiments, we used Google as our primary search-engine, and Yahoo! Search as our secondary search engine. We then repeated the process outlined in sections I and II to train a decision-tree classifier on the data-set. Table 3 shows the precision and recall of the classifier, while Figure 7 shows the actual rule learnt.

Section IV: Error Analysis

Though the classifier built was able to recall documents of the positive class with good accuracy, it seemed to not able to distinguish the negative class examples from the positive ones. We studied the misclassified negative class documents, and found that they were indeed a part of Google's result-set, but ranked below position 20. The average rank of the misclassified negative class documents in the Google result-set was 34. This could explain to some extent, why the classifier misclassified it in the first place.

Section V: Interpretation of learnt rules

Studying the rules output by the decision-tree gives us some insights into what the rules used by commercial web-search engines could be. It seems apparent, in all our experiments, that query-dependent features like matches in various fields override query-independent features like the website's linkage (and as a consequence, perhaps PageRank). This makes intuitive sense – having a high PageRank does not mean that the page is good for *all* queries.

Furthermore, matches in different contexts are weighed differently, with the title of the web-page being the most important to the actual text of the web-page being least important. Anchor-text seems to be important too, but as noted by the anchor-text-numocc feature, a large number of votes need to be cast in terms of referrals for the anchor-text to be significant.

The decision rule learnt by training on the Google-Yahoo! data-set from Section III is similar in spirit to those learnt from section II. An interesting insight into the differences between Google and Yahoo! Search however, seems to be that Google is stricter on pages that have a lot of in-links but otherwise don't match the query (possibly to counter spam-pages that harvest links).

Conclusion

In this paper, we analyzed features used by commercial web search-engines, and their effects on the computed relevance. We were able to establish relationships between some of these features and the rank bestowed upon them.

We also built a model of these features which reflects the underlying model used by these commercial web search-engines. In doing so, we were also able to establish that query-matching features were deemed more important than query-independent features like the link-attributes, and that matches in fields like the title were more important than matches in fields like the page-text.

The difference in two commercial web-search engines (Google and Yahoo! Search) hinted that thought the two search engines produce quite similar result-sets, Google may be fighting spam-pages which use link-harvesting harder than Yahoo! Search is.

References

[1] *http://www.google.com*
[2] *http://www.search.yahoo.com*
[3] *http://www.live.com/*
[4] *'Google Keeps Tweaking Its Search Engine' - New York Times, June 3rd 2007*
[5] *http://www.seomoz.org/article/search-ranking-factors*
[6] *http://www.google.com/support/webmasters/bin/answer.py?answer=35769&hl=en*
[7] *http://www.sempo.org/learning_center/research*
[8] *http://buzz.yahoo.com*
[9] *http://www.dmoz.org*
[10] *http://sitexpolorer.search.yahoo.com*
[11] *Ding et al. Link analysis: hubs and authorities on the world wide web*
[12] *Pandurangan et al. Using PageRank to Characterize Web Structure*
[13] *Friedman, J. H. "Stochastic Gradient Boosting ." (Feb. 1999a)*
[14] *Friedman, J. H. "Greedy Function Approximation: A Gradient Boosting Machine." (March 1999b)*
[15] *Zhu, Ji; Rosset, Saharon; Zou Hui; Hastie, Trevor "Multi-class Adaboost" (Jan. 2006)*
[16] *Ping Li, Christopher J.C. Burges, QiangWu "Learning to rank using classification and gradient boosting"*
[17] *Zhaohui Zheng, Hongyuan Zha, Tong Zhang, Olivier Chapelle, Keke Chen, Gordon Sun "A General Boosting Method and its Application to Learning Ranking Functions for Web Search"*