CS276: Programming Assignment 4

Raj Bandyopadhyay (rajb2), Rafael Guerrero-Platero (rplatero)

May 29, 2013

Introduction

In this assignment, we use two important machine learning techniques, Linear Regression and Support Vector Machines, to automatically rank query results by relevance. As part of using machine learning, we design features to improve the relevance ranking as compared to a set of human-assigned rankings.

Overall results

Table 1 provides an overall summary of the results we have obtained. In the rest of this document, we shall describe our methodology and findings in more detail.

Task 1

As per the description of Task 1, we express each query-page combination as a single vector of five features. Using the Linear Regression implementation in scikit-learn, we fit a model to the training data and use it to evaluate the development data.

Table 2 shows the weights from PA3 (using cosine similarity) and the results we achieved. It's clear that the machine-learned parameters provide a much better performance in ranking. The most surprising result for us was the weights assigned to the body and anchor fields. It is clear from the performance of linear regression that our estimates of their importance were well off the mark in PA3.

Choice of scaling

We experimented with both raw and scaled vectors. Based on the NDCG score improvements we obtained, we settled on the following choices for all our experiments:

- *Log-scaling for Document term vectors:* Log scaling for document TF vectors gives us a significant jump in NDCG scores (1.4%)
- *Length normalization for the 'Body' field:* We experimented with length normalization for different fields, however only the one for the body field gave us a small improvement (0.1%). This is probably because only the body has significant differences in lengths for different documents.

Table 1: Summary of results

Task	Method	Best NDCG score
1	Pointwise Linear Regression	0.8622
2	Pairwise SVM	0.8700
3	Pairwise SVM with extra features	0.8950
Extra	Pointwise Linear Regression with L1-regularization (Lasso)	0.8633

Table 2: Comparison of weights from PA3 and using Linear Regression

	url	title	header	body	anchor		NDCG	
PA3 (manual)	0.553	0.498	0.442	0.498	0.055		0.835	
Linear Regression	0.438	0.695	0.284	0.136	0.475		0.862	

Table 3: Comparison of model weights using Linear Regression vs SVM

	url	title	header	body	anchor		NDCG	
Linear Regression	0.438	0.695	0.284	0.136	0.475		0.862	
Support Vector Machines	0.412	0.101	0.349	0.181	0.813		0.870	

- *IDF for Query term vectors:* We assume that each term occurs only once in most queries, so using term frequency would be redundant. However, using the IDF as the value for each query term does improve NDCG score slightly (0.4%).

Task 2

In this task, we train an SVM using a pairwise method as described in the assignment. We calculate scaled feature vectors from the training data similar to Linear Regression. However, this task requires the extra step of *standardization*. We then calculate the differences between pairs of vectors corresponding to the same query and use that to train the SVM. One interesting fact is that using log-scaling for document term vectors gives us almost 2% improvement in NDCG scores, most likely because SVMs are easily skewed by high variability in feature values. This is the same reason that we perform standardization.

Comparison of results

Tables 4 and 5 show us the top results for three queries using Linear Regression and SVM respectively. The main difference is striking: Linear Regression tends to prefer documents with one or more of the query terms in the URL, whereas SVM identifies the information need better.

The weights in Table 3 suggest why this might be occurring. SVM weights the anchor much more heavily than any other field, thereby emphasizing other pages pointing at this page with a certain anchor text. This serves as a proxy for identifying information need. In fact, assigning a higher weight to the anchor text can be thought of as a substitute for PageRank. In contrast, Linear Regression places a high weight on the title of the page, which may not always be the best guide to the user's information need.

Some observations

On studying the query results in depth, we notice a few things which suggest potential extra features for the next task:

Table 4: Top result for three queries using Linear Regression

Query	Top Result
nvidia auditorium	`http://energyseminar.stanford.edu/nvidia`
sunet login	`http://www.stanford.edu/services/sunetid/`
cs276 stanford	`http://scpd.stanford.edu/search/publicCourseSearchDetails.do?method=load&courseId=11807`

Table 5: Top result for three queries using SVM

Query	Top Result
nvidia auditorium	http://campus-map.stanford.edu/
sunet login	https://accounts.stanford.edu/
cs276 stanford	http://cs276.stanford.edu/

- For similarly relevant URLs, shorter URLs are usually better (`http://cs276.stanford.edu` is better than `http://www.stanford.edu/classes/cs276`)
- Static URLs are usually more relevant than dynamic URLs. Dynamic URLs tend to be longer and contain characters such as '?'.
- The earlier a query term occurs as part of a URL, the more relevant it's likely to be (same example as above).

In general, these observations reflect that given two documents which appear equally relevant, we notice that the user is looking for the more *general* document. This is perhaps because the general document is more likely to contain relevant links to other, more specific information. In the next section, we use these observations to create useful features.

Task 3

In this section, we improve SVM-based ranking using extra features, achieving an NDCG score as high as 0.895. We first implement the features suggested in the assignment i.e. BM25F score, PDF, PageRank and Small window size. Individually, all of them improve the NDCG score, but not necessarily in all combinations. In the case of the small window size feature, only the value for the 'body' field serves to improve the score, most likely because a window does not really carry much meaning in the other fields.

Additional features

We implement a few other features based on our observations in Task 2:

- Number of tokens in URL and Title: Both of these reflect the idea that for similarly relevant documents, shorter URLs or Titles are better since they are more general.
- Presence of '?' character in URL to indicate dynamic pages.
- Position of query terms in URL: For example, if the query contains the term "cs276", a URL containing "cs276" closer to the beginning will be preferred. We simply add up the positions of each query token that occurs in the URL and use that sum as a feature.

Results

Table 6 shows the cumulative effect of each feature. Some notable observations:

- Detecting if the document is a PDF has a very significant effect, followed by the URL length (number of tokens).

Table 6: Benefits of extra features (each row shows the cumulative improvement for all the previous features)

Feature description	NDCG score (cumulative)	Model weights				
Basic SVM	0.8700	url	title	header	body	anchor
		0.479	0.167	0.227	0.329	0.375
PDF url	0.8779	0.069				
PageRank	0.8741	0.018				
Small Window (body only)	0.8724	0.06				
BM25F	0.8754	0.492				
Number of tokens in url	0.8933	-0.226				
Number of tokens in title	0.8927	-0.208				
Url has a '?' character	0.8950	0.041				
Position of query terms in url	0.8861	-0.3				
Best combination of extra features	0.8950					

Table 7: Comparison of model weights using Linear Regression vs Lasso Regression

	url	title	header	body	anchor		NDCG	
Linear Regression	0.438	0.695	0.284	0.136	0.475		0.862	
Lasso with $\alpha = 0.1$	0.225	0.676	0.242	0.159	0.638		0.863	
Lasso with $\alpha = 1$	0	0	0	0.148	0.989		0.867	

- Surprisingly, BM25F and PageRank reduce the NDCG score in this particular ordering of features for this development set, however, the NDCG drops if we remove either of them from the combined feature mix. Hence, we decided to keep them in.
- As expected, the coefficients for number of tokens in URL and Title are negative i.e. the relevance improves with decreased length.
- Some of our suggested features, such as the position of query terms in the URL or the presence of '?' don't improve the performance at all.
- The small window size has a negative coefficient, as expected, indicating that relevance increases with smaller window size.

On the whole, feature selection is a difficult problem, since different features appear to work (or not) in particular combinations with other features. Therefore, a lot of empirical testing and methodologies such as *ablation* are used to identify the best feature sets in practice.

Extra credit (Lasso)

For the extra credit, we implement basic Linear Regression (no extra features) with L1-regularization. We obtain some very intriguing results, as shown in Table 7. As we increase the value of α, the weight assigned to the body remains about the same, while the weights for url, title and header fields moves to the anchor field, increasing the NDCG score at the same time. Our hypothesis is that this reflects the importance of the anchors, and is evidence for why PageRank (for which anchors are a proxy) works so well.