

## CS599: Structure and Dynamics of Networked Information (Spring 2005)

01/26/2005: Text-Based Information Retrieval

Scribes: Viral Shah and Dustin Reishus

Text-based information retrieval has become very important. Search engines must identify web pages most relevant to a user's query out of the billions of pages on the Internet. In previous lectures, we saw how PageRank and HITS are used to identify authoritative pages among all the relevant ones. Here, we turn to the question of how to even identify relevant pages. The most naïve search strategies just return pages that contain the query. There are two problems with this: synonyms (different words having the same meaning) and polysemy (the same word having more than one meaning). Synonyms may lead the search engine to miss relevant pages, because the exact query terms may not appear in all relevant pages. Polysemy may lead the search engine to return irrelevant pages; the pages may contain the search term, but in a different context and with a different meaning than the user intended.

Consider the following table. There are 6 pages, and 6 words occurring on the pages. Page 1 contains words 1, 2, and 3, and so on. Imagine searching these pages for word 3. A naïve search engine would simply return pages 1, 2, and 4, because these pages all contain the query word. However, we might argue that page 3 is also relevant: while it does not contain the exact query term, it is very similar to pages 1 and 2 that do contain the query term. Similarly, page 4 may not be considered relevant: while it does contain the query word, it is very similar to pages 5 and 6 that do not contain the query word.

	Word 1	2	3	4	5	6
Page 1	x	x	x			
2	x	x	x			
3	x	x				
4			x	x	x	x
5				x	x	x
6				x	x	x

To put these observations on a more rigorous and general footing, we can use techniques from Spectral Analysis of Data [1], which is also known as Latent Semantic Analysis [2] in the Information Retrieval community. We consider the table as a matrix, where the cells with an x are 1 and the cells without an x are 0. This matrix is called the *document-term matrix*.

In general, this matrix can be used for representing variety of data sets, where rows index objects in the data set, columns index attributes of those objects, and the  $[i, j]$  entry of the matrix represents the value of the  $j$ -th attribute in the  $i$ -th object. Some examples of interest are where both rows and columns refer to web sites and the  $[i, j]$  entry indicates that site  $i$  has a link to the site  $j$ ; another is that rows index individuals, columns index products, and the  $[i, j]$  entry indicates whether individual  $i$  is interested in, or has previously purchased, product  $j$ .

The main tool in extracting the latent structure from the matrix  $A$  is the singular-value decomposition:

**Lemma 1** *Let  $A$  be an  $m \times n$  matrix (e.g. document-term matrix). Then,  $A$  can be written as  $A = U \cdot \Sigma \cdot V^T$  where  $U$  is an  $m \times k$  orthonormal matrix and  $V$  is an  $n \times k$  orthonormal matrix ( $k = \text{rank}(A)$ ) and*

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \sigma_k \end{bmatrix}$$

*is a  $k \times k$  diagonal matrix with  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k \geq 0$ . This is called the singular value decomposition (SVD) of  $A$ , and the  $\sigma_i$  are called the singular values.*

Let us look more closely at the entries of  $A$ . Entry  $a_{ij} = \sum_{\ell=1}^k u_{i\ell} \cdot \sigma_{\ell} \cdot v_{j\ell}$ . We can consider each  $\ell = 1, \dots, k$  to be a “concept”. Then, we can interpret  $u_{i\ell}$  as “how much is page  $i$  about concept  $\ell$ ”, and  $v_{j\ell}$  as “how much does word  $j$  belong in concept  $\ell$ ”, and  $\sigma_{\ell}$  as the “importance” of concept  $\ell$ . Note that in representing  $A$  in this way, we are making the assumption that concepts behave linearly: the frequency with which a word occurs on a page is the *sum* over all concepts, and there are no superlinear amplifications, or sublinear reductions.

Row  $i$  of  $U$  can be seen as a  $k$ -dimensional description of page  $i$ . Concept  $\ell = 1, \dots, k$  corresponds to columns of  $U$  (or columns of  $V$ ), so the  $k$ -dimensional description of  $i$  expresses it in terms of the concepts. Notice that the same column in  $U$  and  $V$  correspond to the same concept. Also, notice that since  $U$  and  $V$  are orthonormal, concepts are orthogonal.

Following our intuition result, we expect “similar” pages to have similar concept vectors. We can then measure the similarity of two pages by comparing their concept vectors. If there is a small angle between the vectors (or a large inner product) the two pages are about the same (or very similar) concepts. However, up to this point, we are still simply using all the data in the matrix  $A$ : merely rephrasing it in terms of the SVD does not yet give improved search results.

The important issue which we wanted to address was that the entries of the term-document matrix were not derived from an ideal generation process materializing the concepts in the form of terms. Rather, real matrices have “errors”. More formally, if the world were about  $k \ll \min(m, n)$  concepts, then an “ideal world” matrix would have rank  $k$  and thus  $\sigma_{\ell} = 0$  for all  $\ell > k$ . In the real world, web pages don’t conform to our ideal concepts. People write web pages, and their individual tendencies and writing styles vary. This will cause  $A$  to have rank (almost)  $\min(n, m)$ . The concepts  $\ell = k + 1, \dots, \min(n, m)$  are “error-correcting” concepts explaining peculiarities of  $A$ . Hence, to get at the “true” contents of pages, We would like to prune out the entries derived from error-correcting concepts.

How do we do this? First, we determine the “right”  $k$ . Then, we set all  $\sigma_{\ell} = 0$  for  $\ell > k$ . Define

$$\Sigma_k = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \ddots & 0 & \\ & 0 & \sigma_k & 0 \\ \vdots & & 0 & 0 \\ & & & \ddots \\ 0 & \cdots & & 0 \end{bmatrix}$$

then define  $A_k = U \cdot \Sigma_k \cdot V^T$ ,  $U_k = [u_1 \ u_2 \ \cdots \ u_k]$ , and  $V_k = [v_1 \ v_2 \ \cdots \ v_k]$  where the  $u_i$  and  $v_i$  are the columns of  $U$  and  $V$ , respectively. The resulting matrix  $A_k$  is still an  $m \times n$  matrix, but it now has rank  $k$ . The following Lemma shows that it approximates  $A$  well.

**Lemma 2** *The matrix  $A_k$  is the best rank- $k$  approximation to  $A$ : it minimizes  $\|A - B\|_2$  over  $B$  of rank  $k$ , where  $\|A - B\|_2 = \max_{\|x\|_2=1} \|(A - B) \cdot x\|_2$ .*

Given a document-term matrix  $A$  and a query  $q$ , we would like to find the pages in  $A$  that are most relevant to  $q$ . First, we compute  $A_k$  (for some choice of  $k$ ). We consider the space  $\mathbb{R}^k$  as the “concept space”, into which we can map both words and pages based on their rows in  $U$  and  $V$ . Specifically, each page  $i$  is identified with the  $i^{\text{th}}$  row of  $U_k$ , and each word  $j$  with the  $j^{\text{th}}$  row of  $V_k$ . To find pages relevant to the query, we simply output the pages closest to  $q$  in concept space.

How do we choose a good  $k$ ? A good  $k$  is one with  $\sigma_k \gg \sigma_{k+1}$ . If no such  $k$  exists, then  $A$  cannot be approximated well with low rank, although we will still want to choose some  $k$  — it just does not come with good guarantees.

The result of this approach is that we can (hopefully) extract meaningful concepts, and thus identify two pages as similar, and relevant to a query, even if the terms occurring in them are quite different. Conversely, even if a search term occurs in a document, we may be able to filter it out as irrelevant if other terms force it to lie far away in concept space from the point at which our query is located.

Similar ideas can be applied to several other scenarios:

- Collaborative Filtering (or Recommendation Systems) - the reconstruction of the missing data items. Finding similarities between users (again, in some “concept space”), lets us predict which unseen items a user might like.
- Determining the relative importance of documents used to cluster documents in the area of interest.
- Web site ranking.
- Shopping Predictions.

## References

- [1] Y. Azar, A. Fiat, A. Karlin, F. McSherry, and J. Saia. Spectral analysis of data. In *Proc. 33rd ACM Symp. on Theory of Computing*, pages 619–626, 2001.
- [2] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R. Harshman. Indexing by latent semantic analysis. *J. of the American Society for Information Sciences*, 41:391–407, 1990.