Intro to vector space model and latent semantic indexing

Ella Bingham
HUT Lab of Computer and Information Science
ella@iki.fi

September 17, 2003

Vector space model

Related to all text analysis, not just web mining

Nice and fast numerical methods exist. How can they be used with text documents?

Text docs must be somehow presented as numbers. The vector space model is an efficient way to do this.

Bag of words: present a doc as a collection of words, forgetting about the order of the words in the document.

Present a doc as a vector in a high dimensional space where each dimension corresponds to a different word. Number of different words in a large doc corpus is huge $\longrightarrow$ need to select a smaller number of words

- **stemming:** take away the endings. Examples: say, said $\longrightarrow$ sai
  course, courses $\longrightarrow$ course
- **stopwords:** some words do not characterize the document, so take them away. Examples: and, the, is, at
- remove words that occur very seldom in the corpus

The above steps give us the **vocabulary**

Now the j-th entry of a document vector tells something about the appearance of the j-th term in the vocabulary:

- how many times the term appears in the document
- 1 if it appears, 0 if it does not
- a function such as the **term weight:** characterizes the importance of the term
  - **term frequency** (TF) $\frac{\text{how many times term } j \text{ appears in doc } i}{\text{length of doc } i}$
  - **inverse document frequency** (IDF) $\log \frac{\text{total number of docs}}{\text{how many docs contain term } j}$
  - compute TF times IDF (or other function of them)

Collect doc vectors into a **term by document matrix**

Document similarity

Documents are similar $\Leftrightarrow$ their doc vectors are similar

Similarity of vectors often measured as the **cosine** of the angle between them:

$$\cos(x, y) = \frac{x^T y}{||x|| \cdot ||y||}$$

where x, y are doc vectors. This is also a normalized correlation coefficient between the vectors.

Doc vectors often sparse $\longrightarrow$ savings in computations

Queries

Assume you have a doc corpus as a term by doc matrix.

Present the query in the same vector space as the corpus (ignore query terms that do not belong to the vocabulary of the corpus).

Compute the similarity between the query and all docs in the corpus. Return the most similar (that is, best matching) docs.

Was it successful?

- Assume that a "correct answer" to the query exists, that is, there is a set of docs R^* that is **relevant** to the query
- The set of docs R returned by the system are called **retrieved**
- **Precision:** what percentage of retrieved docs are relevant? $\frac{\text{how many in } R \text{ belong to } R^*}{\text{size of } R}$
- **Recall:** what percentage of all relevant docs are retrieved? $\frac{\text{how many in } R \text{ belong to } R^*}{\text{size of } R^*}$
- a tradeoff must be made!

Latent semantic indexing

Problems: **synonymy** (car, automobile) and **polysemy** (bank). The meaning of the term is not always clear. Can be clarified by looking at other terms in the doc, or the appearance of the term in other docs.

A popular method for revealing the true meaning is **latent semantic indexing**: Compute the singular value decomposition (SVD) of the term by doc matrix X

$$X \approx X_K = U_K S_K V_K^T$$

where K largest singular values are computed. So we are discarding the information in the smallest singular

values.

SVD is efficient if X sparse and K small compared to the original dimensionality of X.

SVD gives a set of uncorrelated, latent indexing variables both for docs (V) and for terms (U). Each term or doc is presented by its vector of latent variables. Now the similarity of docs (or terms) is measured by the similarity of their latent variables. Suitable for queries, too.

A problem of SVD/LSI is that latent variables can be negative and thus difficult to interpret. Newer matrix factorization methods such as PLSA, NMF and ICA overcome this problem and sometimes give a more meaningful representation.

Useful papers etc

BOW toolkit for creating term by doc matrices and other: http://www.cs.cmu.edu/~mccallum/bow

Scott Deerwester et al: Indexing by latent semantic analysis, Journal of the American Society for Information Science, Vol 41, No 6, pp. 391–407, 1990. ("The" LSI paper)

Michael Berry et al: Computational methods for intelligent information access, 1995 (Contains LSI updating)

Parry Husbands et al: On the use of singular value decomposition for text retrieval, 2001

Chris Ding: A similarity-based probability model for latent semantic indexing, Proc. of 22nd ACM SIGIR'99 Conference, pp.59–65

Chris Ding: A probabilistic model for dimensionality reduction in information retrieval and filtering, 2001

Exercise

Analyze the term by document matrix of newsgroup data and the corresponding term list (Anne has the data). There are 100 terms and almost 100 documents from each of the newsgroups soc.religion.christian, sci.med, sci.space and sci.cryptography; 348 docs in total. Compute the LSI of the matrix with $K = 10$.
a) List the most important words for each latent dimension.
b) Plot the "activity" of each latent dimension for each document (that is, $K = 10$ signals each of length 348). What do you see?