Project Discussion

Examples of Good projects

1. Something new (e.g. new algorithm, new approach, software).
2. Comprehensive comparison between different existing approaches.
3. Build a great software system.

Announcements

There is a new course newsgroup, *utexas.class.cs395t-datamining*. It's purpose is to act as a forum to discuss project.

SVD for Query Retrieval

Review

Let $\boldsymbol{A}$ is a word by document matrix and q be q query vector.
Note: "word" can be replaces by other features (e.g. "term", "phrase").

Instead of $\boldsymbol{A^t q}$ (exact keyword matching) calculate $\boldsymbol{A^t{}_k q} = \boldsymbol{(V_k \Sigma_k)(U^t{}_k q)}$

- $\boldsymbol{U^t{}_k q}$ is the orthogonal projection of query vector onto lower dimensional space span($\boldsymbol{u_1}, \boldsymbol{u_2}, \ldots \boldsymbol{u_k}$).
- Note the reduction in dimension of q from w to k.

Similarly, $\boldsymbol{V_k \Sigma_k}$ can be interpreted as a projection of all documents onto span($\boldsymbol{u_1}, \boldsymbol{u_2}, \ldots \boldsymbol{u_k}$).

- Note that $\boldsymbol{U^t{}_k A = \Sigma_k V^t{}_k}$

Why does Ak work better than A?

Last time I gave the argument given by LSI authors: $\boldsymbol{A_k}$ contains "signal" and the "noise" $\boldsymbol{A} - \boldsymbol{A_k}$ is removed.

Another argument is that : Instead of $\boldsymbol{A}$, the writer implied a slightly different ideal matrix ***A(ideal)*** which would be obtained from $\boldsymbol{A}$ by incorporating "synonymy" and "polysemy". We can think of the k-truncated SVD as an approximation to ***A(ideal).***

LSI Example:

An example of a document collection was presented in class.

The collection consisted of two different types of documents.

The SVD gives a mechanism to look at the co-occurrence matrix.

- We form $\boldsymbol{A^t A}$ (the co-occurrence matrix) and look at its graph.

- Co-occurrences (different words occurring in the same document simultaneously) are 1–2 hops away

Note: There are 2 ways to form ***A***.

1. Form document vectors based solely on word occurrence
2. Do step 1, but normalize the document vectors to lie on the unit sphere.

The class example used SVD decomposition with k=2 for query retrieval.

- The first singular vector had the largest entries in positions associated with the most frequently occurring words. One would expect this because the first singular vector is supposed to be the best rank 1 approximation to ***A***.

- The documents projected onto the two singular vectors of the example seemed to form two distinct clusters.
- The clusters became even more defined when normalized document vectors were used.

Clustering methods

- Given projections of docs onto the first k singular vectors, find lines (or hyperplanes for k>2) that give the maximum separation of document projections

Terms used in query retrieval experiments

- Precision and Recall

Both are used as human calculated measurements that indicate how well a given query "performed" in a collection of documents

Drawbacks of SVD/LSI

1. Computationally expensive
    - I. Matrix is large and sparse.
    - II. Typically, many singular vectors required (k = 100 – 500)
2. Not very intuitive. How does one quantify the statement "Singular vectors capture concepts of a document collection"?
3. While LSI seems to work well for long queries, people have found that it is not really good for short queries (as would typically be seen on the internet). Thus, not used in any commercial engine
4. Fundamental problems with vector space model (e.g. You might never want to return a page containing only "Bill Clinton Sucks." However, the vector space model allows no way to automatically specify this.)