Problem2-Report

April 23, 2020

Ziqi Tan U 88387934

1 SVD and LSI

1.1 Data preprocessing

1. Change all letters to lower case.
2. Replace all punctuations by a space.
3. Remove all the numbers if necessary.

1.2 Stop words

Remove all the stop words.

1.3 Extract the stems of each word

Replace all the words by their stem words.

1.4 Construct a term-document matrix A1

Suppose we have N documents and n terms in all. Then we will get a n by N matrix A1.

1.5 Construct a tf-idf matrix A2

Suppose we have N documents and n terms in all. Then we will get a n by N matrix A2.

1.6 Verterize the query

From all documents, we can get a dictionary. The query should be vectorized as a column of matrix A1. Let the query vector be q.

1.7 SVD Decomposition

1. Perform singular value decomposition on matrix A.

$$A = USV^T$$

Rows of V holds right eigenvectors. S is a diagonal matrix, the elements of whose diagonal hold the singular value. Columns of U holds left eigenvector.

2. Implement a Rank k Approximation by keeping the first two columns of U and V and the first two columns and rows of S.

$$U \leftarrow U_k$$

$$S \leftarrow S_k$$

$$V^T \leftarrow V_k^T$$

3. Now vector

$$V_k = [d_1, d_2, ..., d_N]$$

holds the information of N documents.

We make the query q become the same thing as d_i.

$$q = q^T U_k S_k^{-1}$$

1.8 Query

1. Normalize q and d_i.
2. Calculate the cosine similiarity between q and d_i.
3. Report the top most similar documents.

2 Result

Queries: [‘music play compos’, ‘ancient univers island’, ‘patient friend’, ‘loan pay money’, ‘work wealth fun’, ‘cold rain fog’]

- preverse first 25 eigenvectors
  - preserve 82.0% information
  - Query by Matrix A1:
  - [[27, 25, 26], [20, 2, 4], [12, 15, 6], [23, 14, 19], [1, 7, 27], [33, 7, 28]]
- Query by Matrix TF-IDF:
  - preverse first 27 eigenvectors
  - preserve 80.0% information
  - [[27, 25, 26], [20, 35, 2], [15, 12, 6], [23, 3, 14], [32, 27, 7], [34, 31, 33]]