Latent Semantic Indexing
Lecture 8: October 4, 2013

CS886-2 Natural Language Understanding

University of Waterloo

Vector Space Model

- Idea:
  - Treat words as features
  - Count frequency of each word in a document
- Problem:
  - Synonyms: VSM does not merge words with same meaning
  - Polysemy: VSM does not distinguish different meanings of the same word

Latent Semantics

- Can we consider different features than words to represent documents?
- Yes, obtain a new latent space by matrix decomposition

Co-occurrence matrices

- Let C be the term-document matrix such that
  $C_{ij} = \begin{cases} 1 & if\ term\ i\ appears\ in\ document\ j \\ 0 & otherwise \end{cases}$
- Let $C^T C$ be a document-document matrix such that $(C^T C)_{jj'} = n$ indicates that documents j, j' have n common words
- Let CC^T be a word-word matrix such that $(CC^T)_{ii'} = n$ indicates that words i, i' co-occur in n documents.

Matrix Decompositions

- Singular Value Decomposition

$$C = U\Sigma V^T$$

- Eigen-decompositions:

$$CC^T = U\Sigma^2 U^T$$
$$C^T C = V\Sigma^2 V^T$$

Interpretation

- Rows of U: new word representation
- Rows of V: new document representation

Low Rank Approximation

- Goal: find the best basis of k dimensions that approximates C
- Interpret this reduced basis as some kind of semantic latent space

- Solution: Minimize Frobenius norm

$$\min_{\{Z|rank(Z)=k\}} ||Z - C||_F = ||U_k \Sigma_k V_k^T - U \Sigma V^T||_F = \sigma_{k+1}$$

- Where Σ_k, U_k, V_k are truncated versions of Σ, U, V for the k largest singular values

Embedding queries

- Queries to rank relevant documents can be computed by $q^T d$

- Embed query q and document d in latent space by computing $q_k^T d_k$ where
  - $q_k = \Sigma_k^{-1} U_k^T q$
  - $d_k = \Sigma_k^{-1} U_k^T d$

Problems

- Latent space difficult to interpret
  - Especially negative numbers
- SVD is time consuming