CS657A: Information Retrieval
Term Embedding Models

Arnab Bhattacharya
arnabb@cse.iitk.ac.in

Computer Science and Engineering,
Indian Institute of Technology, Kanpur
http://web.cse.iitk.ac.in/~cs657/

2nd semester, 2021-22

Tue 1030-1145, Thu 1200-1315

Embedding Models

- Models for *embedding* terms to vector spaces
- *Term-document* matrix
  - Rows are terms, columns are documents
  - Values generally indicate counts
- *Term-term co-occurrence* matrix
  - Whether terms co-occur in a document
  - Tries to capture context
- *Term-term context* matrix
  - May use a special context window
  - Whether occurs within k terms upstream and downstream

Singular Value Decomposition (SVD)

- Singular value decomposition is factorization of a matrix

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

- If A is of size $m \times n$, then U is $m \times m$, V is $n \times n$ and Σ is $m \times n$
- Columns of U are *eigenvectors* of AA^T
  - Left singular vectors
  - $UU^T = I_m$ (orthonormal)
- Columns of V are *eigenvectors* of A^TA
  - Right singular vectors
  - $V^TV = I_n$ (orthonormal)
- σ_{ii} are the singular values
  - Σ is *diagonal*
  - Singular values are *positive square roots* of *eigenvalues* of AA^T or A^TA
- $\sigma_{11} \geq \sigma_{22} \geq \cdots \geq \sigma_{nn}$ (assuming n singular values)

Transformation using SVD

- Transformed data

$$T = AV = U\Sigma$$

- V is called SVD transform matrix
- Essentially, T is just a rotation of A
- Dimensionality of T is n
- n different basis vectors than the original space
- Columns of V give the basis vectors in rotated space
- V shows how each *document* can be represented as a linear combination of other documents
- U shows how each *term* can be represented as a linear combination of other terms
- Lengths of vectors are preserved

$$||\vec{a_i}||_2 = ||\vec{t_i}||_2$$

Example

$$A \begin{bmatrix} 2 & 4 & 1 \\ 1 & 3 & 0 \\ 5 & 2 & 1 \\ 0 & 0 & 7 \\ 3 & 3 & 3 \end{bmatrix} = U \begin{bmatrix} -0.41 & 0.29 & 0.49 & -0.41 & -0.56 \\ -0.23 & 0.27 & 0.48 & 0.77 & 0.18 \\ -0.48 & 0.36 & -0.71 & 0.23 & -0.25 \\ -0.47 & -0.83 & 0.02 & 0.18 & -0.19 \\ -0.55 & 0.05 & 0.01 & -0.37 & 0.73 \end{bmatrix}$$

$$\times \Sigma \begin{bmatrix} 9.30 & 0 & 0 \\ 0 & 6.47 & 0 \\ 0 & 0 & 2.91 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times V^T \begin{bmatrix} -0.55 & 0.44 & -0.70 \\ -0.53 & 0.45 & 0.70 \\ -0.63 & -0.77 & 0.01 \end{bmatrix}^T$$

Transformed Data

$$T = AV = U\Sigma = \begin{bmatrix} 2 & 4 & 1 \\ 1 & 3 & 0 \\ 5 & 2 & 1 \\ 0 & 0 & 7 \\ 3 & 3 & 3 \end{bmatrix} \times \begin{bmatrix} -0.55 & 0.44 & -0.70 \\ -0.53 & 0.45 & 0.70 \\ -0.63 & -0.77 & 0.01 \end{bmatrix}$$

$$= \begin{bmatrix} -3.89 & 1.93 & 1.44 \\ -2.16 & 1.80 & 1.42 \\ -4.47 & 2.36 & -2.08 \\ -4.45 & -5.39 & 0.08 \\ -5.18 & 0.38 & 0.05 \end{bmatrix}$$

$$\text{Lengths} = [4.58, 3.16, 5.47, 7.00, 5.19]$$

Compact Form

$$A \begin{bmatrix} 2 & 4 & 1 \\ 1 & 3 & 0 \\ 5 & 2 & 1 \\ 0 & 0 & 7 \\ 3 & 3 & 3 \end{bmatrix} = U \begin{bmatrix} -0.41 & 0.29 & 0.49 \\ -0.23 & 0.27 & 0.48 \\ -0.48 & 0.36 & -0.71 \\ -0.47 & -0.83 & 0.02 \\ -0.55 & 0.05 & 0.01 \end{bmatrix}$$

$$\times \Sigma \begin{bmatrix} 9.30 & 0 & 0 \\ 0 & 6.47 & 0 \\ 0 & 0 & 2.91 \end{bmatrix} \times V^T \begin{bmatrix} -0.55 & 0.44 & -0.70 \\ -0.53 & 0.45 & 0.70 \\ -0.63 & -0.77 & 0.01 \end{bmatrix}^T$$

- If A is of size $m \times n$, then U is $m \times n$, V is $n \times n$ and Σ is $n \times n$
- Works because there at most n non-zero singular values in Σ

Dimensionality Reduction using SVD

$$A = U \Sigma V^T = \sum_{i=1}^{n} (u_i \sigma_{ii} v_i^T)$$

- Use only k dimensions
- Retain first k columns for U and V and first k values for Σ
- First k columns of V give the basis vectors in reduced space

$$A_k \approx \sum_{i=1}^{k} (u_i \sigma_{ii} v_i^T) = U_{1\ldots k} \Sigma_{1\ldots k} V_{1\ldots k}^T$$

$$T_k \approx A V_{1\ldots k}$$

Reduced Dimensionality

$$A \approx A_k = U_k \begin{bmatrix} -0.41 & 0.29 \\ -0.23 & 0.27 \\ -0.48 & 0.36 \\ -0.47 & -0.83 \\ -0.55 & 0.05 \end{bmatrix} \times \Sigma \begin{bmatrix} 9.30 & 0 \\ 0 & 6.47 \end{bmatrix} \times V^T \begin{bmatrix} -0.55 & 0.44 \\ -0.53 & 0.45 \\ -0.63 & -0.77 \end{bmatrix}^T$$

$$= \begin{bmatrix} 3.01 & 2.97 & 0.98 \\ 2.00 & 1.98 & -0.01 \\ 3.52 & 3.48 & 1.02 \\ 0.05 & -0.05 & 6.99 \\ 3.03 & 2.96 & 2.99 \end{bmatrix}$$

$$T \approx T_k = AV_k = U_k\Sigma_k = \begin{bmatrix} -3.89 & 1.93 \\ -2.16 & 1.80 \\ -4.47 & 2.36 \\ -4.45 & -5.39 \\ -5.18 & 0.38 \end{bmatrix}$$

$$\text{Reduced Lengths} = [4.34, 2.82, 5.06, 6.99, 5.19]$$

$$\text{Length Ratios} = [0.95, 0.89, 0.92, 1.00, 1.00]$$

Best Approximation

- Frobenius norm of a matrix C of size $n \times m$ is

$$||C||_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} C_{ij}^2}$$

- Consider any rank-k approximation A_k of A
- SVD produces A_k^* that minimizes the Frobenius norm of the difference
  - Best in terms of sum squared error

$$A_k^* = \arg \min_{A_k : rank = k} ||A - A_k||_F$$

Latent Semantic Analysis (LSA)

- Applying SVD to term-document matrix is called latent semantic analysis (LSA) or latent semantic indexing (LSI)
- Which lower dimensionality to retain?
- Concept of energy
- Total energy is sum of *squares* of singular values (also called spread or *variance*)

$$E = \sum_{i=1}^{n} \sigma_{ii}^2$$

- Retain k dimensions such that $p\,\%$ of the energy is retained

$$E_k = \sum_{i=1}^{k} \sigma_{ii}^2 \quad \text{s.t.} \quad \frac{E_k}{E} \geq p$$

- Generally, p is between $80\,\%$ to $95\,\%$
- In the example, $k = 1$ ($k = 2$) retains 63% (94%) of energy

Discussion

- LSA with around 100-300 dimensions seem to handle *synonymy* well
  - In English
- SVD is too time-consuming and space-hungry
  - Running time is $O(m \cdot n \cdot r)$ for A of size $m \times n$ and rank r
- Does not capture sequence of terms or context

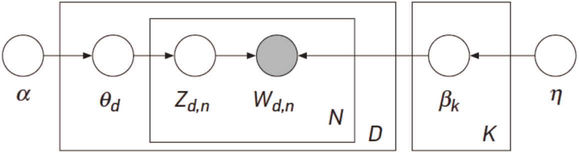
Topics

- He swung the ball more than other fast bowlers
- During lunch break, he eats only bananas
- He was caught chewing the seam of the ball
- To generate more swing, he put a chewing gum
- After his retirement, he ate a lot of chocolates
- Topic 1: cricket; Topic 2: food
- Sentence 1: 100% cricket, 0% food
- Sentence 2: 10% cricket, 90% food
- Sentence 3: 50% cricket, 50% food
- Sentence 4: 30% cricket, 70% food
- Sentence 5: 0% cricket, 100% food

LDA

- Latent Dirichlet Allocation (LDA)
- *Generative probabilistic* model that assumes
  - Each document is a *mixture of topics*
  - Each topic has a *probability distribution of words*
  - Thus, each document is a *mixture of distributions of words*
- How to generate a *corpus* of documents?
  - Generate D documents independently
- Repeat D times (documents)
  - Pick a certain length, i.e., number of words N from some distribution $P(N)$ (Poisson)
  - Decide on the multinomial topic distribution $P(Z|D = d)$ involving K topics
- Repeat N times (words)
  - Pick a topic z according to the chosen topic distribution $P(Z|D = d)$
  - Pick a word $P(w|Z = z)$ from the word distribution $P(W|Z = z)$ corresponding to the topic z
- Latent variables (topics) coming from *Dirichlet* priors *allocate* words

Detailed Model: Plate Notation



- Parameters
  - η: Dirichlet prior on per-topic word distribution
  - β_k: Word distribution for topic k
  - α: Dirichlet prior on per-document topic distribution
  - θ_d: Topic distribution for document d
  - $z_{d,n}$: Topic for n^{th} word in d^{th} document
  - $w_{d,n}$: n^{th} word in d^{th} document
- Generation
  - Choose $\beta_k \sim Dir(\eta)$ for $k = 1, \ldots, K$
  - Choose $\theta_d \sim Dir(\alpha)$ for $d = 1, \ldots, D$
  - For each word position d, n for $d = 1, \ldots, D$ and $j = 1, \ldots, N_d$
    - Choose topic $z_{d,n} \sim Multinomial(\theta_d)$
    - Choose word $w_{d,n} \sim Multinomial(\beta_{z_{d,n}})$

Discussion

- Learning LDA involves finding the parameters that *maximize* the *joint probability* of words, topics, and documents in the corpus

$$p(\beta_K, \theta_D, z_{D,N}, w_{D,N}|\alpha, \eta) = \prod_{k=1}^{K} p(\beta_k|\eta). \prod_{d=1}^{D} p(\theta_d|\alpha).$$
$$\left(\prod_{j=1}^{N_d} \sum_{z_d=1}^{K} p(z_{d,j}|\theta_d).p(w_{d,j}|\beta_k, z_{d,j}) \right)$$

- Can use expectation-maximization (EM)
  - Slow convergence
- Better is to use Gibbs sampling
  - Sample one dimension at a time holding the other dimensions constant
- Around 100-300 topics give better results
- More interpretable

Term Co-occurrence and Context

- Global Vectors (GloVe)
- Global co-occurrence matrix X
- X_{ij} encodes how many times term i has appeared in the context of term j
- Sum of a row $X_i = \sum_{\forall j} X_{ij}$ denotes the *total* number of occurrences of term i
- *Probability* that term j occurs in the context of term i is

$$P_{ij} = P(j|i) = \frac{X_{ij}}{X_i}$$

Raw Probabilities and Ratio

- Raw counts X_{ij} or even raw probabilities P_{ij} may not be useful
- Depends a lot on the actual terms i and j
- Also, asymmetric
- *Ratio* is a better indicator of relevance

	$P(k\|ice)$	$P(k\|steam)$	$\frac{P(k\|ice)}{P(k\|steam)}$
k = solid	1.9e-4	2.2e-5	8.636
k = gas	6.6e-5	7.8e-4	0.084
k = water	3.0e-3	2.2e-3	1.363
k = fashion	1.7e-5	1.8e-5	0.944

- "solid" is more relevant to "ice" than "steam"
- "gas" is more relevant to "steam" than "ice"
- "water" is relevant to both "ice" and "steam"
- "fashion" is irrelevant to both "ice" and "steam"

Ratio of Probabilities

- Therefore, whether a term k is more relevant to term i than term j depends on the ratio P_{ik}/P_{jk}
- Hence, for *context vector* $\tilde{w}_k$ and *term vectors* w_i, w_j

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$$

- Vectors are of some dimensionality d
- Since ratio of probabilities is scalar

$$F((w_i - w_j)^T \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$$

- Replacing probabilities by the same functional form

$$F((w_i - w_j)^T \tilde{w}_k) = \frac{F(w_i^T \tilde{w}_k)}{F(w_j^T \tilde{w}_k)}$$

Symmetry

- Functional form solution is $F = exp$
- Since $F(w_i^T \tilde{w}_k) = P_{ik}$

$$w_i^T \tilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i)$$

$$w_i^T \tilde{w}_k + b_i = \log(X_{ik})$$

- Bias b_i encapsulates count of term i
- Symmetric: both terms i and k should be in the context of each other

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik})$$

- To avoid zero count problems

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(1 + X_{ik})$$

Objective Function

- *Objective function* is weighted with (a function of) X_{ij}

$$\arg\min_{w_i, w_j, \ldots} \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2$$

- Properties of weighting function f
  - $f(0) \to 0$
  - $f(x)$ is non-decreasing
  - $f(x)$ should not increase heavily for large values of x
- Following function is used

$$f(x) = \begin{cases} (\frac{x}{x_{max}})^\alpha & \text{if } x < x_{max} \\ 1 & \text{otherwise} \end{cases}$$

  - $\alpha = 3/4$
  - $x_{max} = 100$

Discussion

- Context window of size ± 5
- Dimensionality of 100 or 300
- Did well on *word analogy* tasks
  - king:man::woman:?
  - Athens:Greece::Berlin:?
  - a:b::c:?
  - Closest vector to $\vec{w}_a - \vec{w}_b + \vec{w}_c$
- Captures local context and global pairwise statistics