Vector Space Model: Latent Semantic Indexing

(Teaching Inspired by Research)

Prof. Dr. Marcin Grzegorzek and
the Medical Data Science Team

UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR MEDIZINISCHE INFORMATIK

Contents of the Course

Week	Lecture	Practical Exercises
1	**(05/04)** Introduction to Medical Information Retrieval (MIR)	**(05/04)** Introduction to Python
2	**(12/04)** Main Components and Classification of MIR Systems	**(12/04)** Introduction to Python
3	**(19/04)** Metadata in Medical Information Retrieval Systems	**(19/04)** CBIR in Medical Applications
4	***(26/04)*** *No Lecture due to a Business Trip*	**(26/04)** CBIR in Medical Applications
5	**(03/05)** Set Theoretic Model: Boolean Retrieval	**(03/05)** CBIR in Medical Applications
6	**(10/05)** Set Theoretic Model: Fuzzy Retrieval	**(10/05)** Flask Tutorial
7	**(17/05)** Vector Space Model: Similarity Measures	**(17/05)** Flask Tutorial

Contents of the Course

8	**(24/05)** Vector Space Model: Distance Functions	**(24/05)** HTML
9	**(31/05)** Vector Space Model: Latent Semantic Indexing	**(31/05)** HTML
10	**(07/06)** Probabilistic Model	**(07/06)** HTML
11	**(14/06)** Text-based Retrieval of Medical Information	**(14/06)** Deep Learning
12	**(21/06)** Audio-based Retrieval of Medical Information	**(21/06)** Deep Learning
13	**(28/06)** Image-based Retrieval of Medical Information	**(28/06)** Relevance Feedback
14	**(05/07)** Demonstrators from Current Research Projects	**(05/07)** Relevance Feedback
15	**(12/07)** Summary and Conclusions	**(12/07)** Evaluation

Synonymy and Polysemy in Information Retrieval

- **Synonymy**: Different words (say *car* and *automobile*) have the same meaning. The vector space representation fails to capture the relationship between synonymous terms such as *car* and *automobile*, because they correspond to separate dimensions in the term-document matrix. Consequently the computed similarity between a query *car* and a document containing both *car* and *automobile* underestimates the true similarity that a user would perceive.

- **Polysemy** refers to the case where a term such as *charge* has multiple meanings, so that the computed similarity overestimates the similarity that a user would perceive.

Transforming the Term-Document Matrix – Example

Consider the following term-document matrix $\boldsymbol{X}$:

	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$	$\boldsymbol{d}_3$	$\boldsymbol{d}_4$	$\boldsymbol{d}_5$	$\boldsymbol{d}_6$
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
voyage	1	0	0	1	1	0
trip	0	0	0	1	0	1

Using the Singular Value Decomposition (SVD, see below), the matrix $\boldsymbol{X}$ can be reformulated as follows:

$$\boldsymbol{X} = \boldsymbol{U}\Sigma\boldsymbol{V}^{\mathrm{T}} \quad .$$

Transforming the Term-Document Matrix – Example

For our example $\boldsymbol{X}$, the matrix $\boldsymbol{U}$ has the following values:

ship	−0.44	−0.30	0.57	0.58	0.25
boat	−0.13	−0.33	−0.59	0.00	0.73
ocean	−0.48	−0.51	−0.37	0.00	−0.61
voyage	−0.70	0.35	0.15	−0.58	0.16
trip	−0.26	0.65	−0.41	0.58	−0.09

Transforming the Term-Document Matrix – Example

For our example $\boldsymbol{X}$, the matrix Σ has the following values:

2.16	0.00	0.00	0.00	0.00
0.00	1.59	0.00	0.00	0.00
0.00	0.00	1.28	0.00	0.00
0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	0.39

Transforming the Term-Document Matrix – Example

For our example $\boldsymbol{X}$, the matrix $\boldsymbol{V}^{\mathrm{T}}$ has the following values:

$\boldsymbol{d}_1$	$\boldsymbol{d}_2$	$\boldsymbol{d}_3$	$\boldsymbol{d}_4$	$\boldsymbol{d}_5$	$\boldsymbol{d}_6$
−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
−0.29	−0.53	−0.19	0.63	0.22	0.41
0.28	−0.75	0.45	−0.20	0.12	−0.33
0.00	0.00	0.58	0.00	−0.58	0.58
−0.53	0.29	0.63	0.19	0.41	−0.22

Transforming the Term-Document Matrix – Example

By “zeroing out” all but the two largest singular values of $\mathbf{\Sigma}$, we obtain $\mathbf{\Sigma}_2$:

2.16	0.00	0.00	0.00	0.00
0.00	1.59	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00

Transforming the Term-Document Matrix – Example

Using Σ_2, we can compute the corresponding version of the term-document matrix $\boldsymbol{X}_2$:

$\hat{\boldsymbol{d}}_1$	$\hat{\boldsymbol{d}}_2$	$\hat{\boldsymbol{d}}_3$	$\hat{\boldsymbol{d}}_4$	$\hat{\boldsymbol{d}}_5$	$\hat{\boldsymbol{d}}_6$
-1.62	-0.60	-0.44	-0.97	-0.70	-0.26
-0.46	-0.84	-0.30	1.00	0.35	0.65

Now, every document is described by a two dimensional vector $\hat{\boldsymbol{d}}_j$. In contrast to the five dimensional vectors $\boldsymbol{d}_j$ of the original matrix $\boldsymbol{X}$, we do not exactly know the semantics behind the dimensions after transformation.

Term-Document Matrix – Notation

$$
\begin{array}{cc}
 & \begin{array}{ccccc} \phantom{x_{1,1}} & & \mathbf{d}_j & & \phantom{x_{1,n}} \\ & & \downarrow & & \end{array} \\
\begin{array}{c} \\ \\ \mathbf{t}_i^T \rightarrow \\ \\ \\ \end{array} & \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,j} & \cdots & x_{m,n} \end{bmatrix}
\end{array}
$$

Term-Document Matrix – Notation

- A row in this matrix is a vector corresponding to a term, giving its relation to each document:

$$\boldsymbol{t}_i = (x_{i,1}, \ldots, x_{i,j}, \ldots, x_{i,n})^{\mathrm{T}} \quad .$$

- A column in this matrix is a vector corresponding to a document, giving its relation to each term:

$$\boldsymbol{d}_j = (d_{1,j}, \ldots, d_{i,j}, \ldots, d_{m,j})^{\mathrm{T}} \quad .$$

Term-Document Matrix – Correlations

- The dot product $\boldsymbol{t}_i{}^{\mathrm{T}}\boldsymbol{t}_p$ gives the correlation between the terms over the set of all documents.
- The matrix product $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ contains all the dot products. The Element (i, p) equal to the element (p, i) contains the dot product $\boldsymbol{t}_i{}^{\mathrm{T}}\boldsymbol{t}_p = \boldsymbol{t}_p{}^{\mathrm{T}}\boldsymbol{t}_i$.
- Likewise, the matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ contains the dot products between all document vectors, giving their correlation over the terms $\boldsymbol{d}_j{}^{\mathrm{T}}\boldsymbol{d}_q = \boldsymbol{d}_q{}^{\mathrm{T}}\boldsymbol{d}_j$.

SVD – Main Statement

From the theory of linear algebra, there exists the following decomposition of the matrix $\boldsymbol{X}$:

$$\boldsymbol{X} = \boldsymbol{U}\Sigma\boldsymbol{V}^{\mathrm{T}}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices and Σ is a diagonal matrix.

SVD – Term and Document Correlations

The term and document correlations can be now reformulated:

$$\boldsymbol{XX}^{\mathrm{T}} = (\boldsymbol{U}\Sigma\boldsymbol{V}^{\mathrm{T}})(\boldsymbol{U}\Sigma\boldsymbol{V}^{\mathrm{T}})^{\mathrm{T}} = (\boldsymbol{U}\Sigma\boldsymbol{V}^{\mathrm{T}})(\boldsymbol{V}\Sigma^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}) = \boldsymbol{U}\Sigma\Sigma^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}$$

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} = (\boldsymbol{U}\Sigma\boldsymbol{V}^{\mathrm{T}})^{\mathrm{T}}(\boldsymbol{U}\Sigma\boldsymbol{V}^{\mathrm{T}}) = (\boldsymbol{V}\Sigma^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})(\boldsymbol{U}\Sigma\boldsymbol{V}^{\mathrm{T}}) = \boldsymbol{V}\Sigma\Sigma^{\mathrm{T}}\boldsymbol{V}^{\mathrm{T}}$$

Since $\Sigma\Sigma^{\mathrm{T}}$ and $\Sigma^{\mathrm{T}}\Sigma$ are diagonal, $\boldsymbol{U}$ must contain the eigenvectors of $\boldsymbol{XX}^{\mathrm{T}}$, while $\boldsymbol{V}$ must be the eigenvectors of $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$.

SVD – Full Notation

$$
\begin{array}{cc}
 & \begin{array}{ccccc} & & X & & \\ & & (\mathbf{d}_j) & & \\ & & \downarrow & & \end{array} \\
(\mathbf{t}_i^T) \rightarrow & \begin{bmatrix} x_{1,1} & \dots & x_{1,j} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \dots & x_{i,j} & \dots & x_{i,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,j} & \dots & x_{m,n} \end{bmatrix}
\end{array}
=
\begin{array}{cc}
 & U \\
(\hat{\mathbf{t}}_i^T) \rightarrow & \begin{bmatrix} \begin{bmatrix} \mathbf{u}_1 \end{bmatrix} \dots \begin{bmatrix} \mathbf{u}_l \end{bmatrix} \end{bmatrix}
\end{array}
\cdot
\begin{array}{c}
\Sigma \\
\begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{bmatrix}
\end{array}
\cdot
\begin{array}{c}
V^T \\ (\hat{\mathbf{d}}_j) \\ \downarrow \\
\begin{bmatrix} [& \mathbf{v}_1 &] \\ & \vdots & \\ [& \mathbf{v}_l &] \end{bmatrix}
\end{array}
$$

- $\sigma_1, \dots, \sigma_l$ – singular values
- $\boldsymbol{u}_1, \dots, \boldsymbol{u}_l$ – left singular vectors
- $\boldsymbol{v}_1, \dots, \boldsymbol{v}_l$ – right singular vectors

SVD – Dimensionality Reduction

- Selecting k largest singular values and their corresponding singular vectors from $\boldsymbol{U}$ and $\boldsymbol{V}$, we get the rank k approximation of $\boldsymbol{X}$ with the smallest error.
- The vector $\hat{\boldsymbol{t}}_i$ is a result of mapping (approximation) the vector $\boldsymbol{t}_i$ into a k-dimensional space.
- The vector $\hat{\boldsymbol{d}}_j$ is a result of mapping (approximation) the vector $\boldsymbol{d}_j$ into a k-dimensional space.
- The full approximation can be expressed as follows:

$$\boldsymbol{X}_k = \boldsymbol{U}_k \Sigma_k \boldsymbol{V}_k{}^{\mathrm{T}} \quad .$$

SVD Applied in Information Retrieval

- First, all documents in the database are transformed into the k-dimensional space using SVD:

$$\hat{\boldsymbol{d}}_j = \Sigma_k^{-1} \boldsymbol{U}_k{}^{\mathrm{T}} \boldsymbol{d}_j \quad .$$

- Then, the query vector $\boldsymbol{q}$ is transformed using the same transformation:

$$\hat{\boldsymbol{q}} = \Sigma_k^{-1} \boldsymbol{U}_k{}^{\mathrm{T}} \boldsymbol{q} \quad .$$

- The inverse of the diagonal matrix Σ_k can be found by inverting each nonzero value within the matrix.

Final Statements

- LSI is a powerful technique to cope with the problems of synonymy and polysemy in information retrieval.
- LSI is able to extract a “non-visible” (latent) semantics from text documents.
- Although LSI reduces the dimensionality of the vector space drastically, it usually leads to a better performance (precision, recall) of information retrieval systems.