Information Retrieval

Weike Pan

The slides are **adapted from those provided by Prof. Hinrich Schütze** at University of Munich (http://www.cis.lmu.de/~hs/teach/14s/ir/).

Chapter 18 Matrix decompositions & latent semantic indexing

Outline

- 18.1 Linear algebra review
- 18.2 Term-document matrices and singular value decompositions
- 18.3 Low-rank approximations
- 18.4 Latent semantic indexing
- 18.5 References and further reading

18.2 Term-document matrices and singular value decompositions

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
anthony	5.25	3.18	0.0	0.0	0.0	0.35
brutus	1.21	6.10	0.0	1.0	0.0	0.0
caesar	8.59	2.54	0.0	1.51	0.25	0.0
calpurnia	0.0	1.54	0.0	0.0	0.0	0.0
cleopatra	2.85	0.0	0.0	0.0	0.0	0.0
mercy	1.51	0.0	1.90	0.12	5.25	0.88
worser	1.37	0.0	0.11	4.15	0.25	1.95
...						

- This matrix is the basis for computing the **similarity** between documents and queries.
- Question: Can we **transform** this matrix, so that we get a better measure of similarity between documents and queries?

18.2 Term-document matrices and singular value decompositions

- We will decompose the term-document matrix into a product of three matrices via singular value decomposition (SVD).

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

C is the term-document matrix

- We will then use the SVD to compute a new and improved term-document matrix *C′*.

- We'll get better similarity values out of *C′* (compared with *C*).

18.2 Term-document matrices and singular value decompositions

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

$=$

U	1	2	3	4	5
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
wood	−0.70	0.35	0.15	−0.58	0.16
tree	−0.26	0.65	−0.41	0.58	−0.09

$\times$

Σ	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	1.28	0.00	0.00
4	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	0.00	0.39

$\times$

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.28	−0.75	0.45	−0.20	0.12	−0.33
4	0.00	0.00	0.58	0.00	−0.58	0.58
5	−0.53	0.29	0.63	0.19	0.41	−0.22

- We use a non-weighted matrix here to simplify the example.

18.2 Term-document matrices and singular value decompositions

U	1	2	3	4	5
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
wood	−0.70	0.35	0.15	−0.58	0.16
tree	−0.26	0.65	−0.41	0.58	−0.09

- One row per term
- U is an orthonormal matrix: (i) Column vectors have unit length. (ii) Any two distinct column vectors are orthogonal to each other.

- Think of the dimensions as “semantic” dimensions that capture distinct topics like politics, sports and economics.
- Each number u_{ij} in the matrix indicates how strongly related term i is to the topic represented by semantic dimension j.

18.2 Term-document matrices and singular value decompositions

Σ	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	1.28	0.00	0.00
4	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	0.00	0.39

- This is a square and diagonal matrix of dimensionality min(M,N) $\times$ min(M,N).
- The diagonal consists of the singular values of C.

- The magnitude of the singular value measures the **importance** of the corresponding semantic dimension.
- We’ll make use of this by omitting unimportant dimensions.

18.2 Term-document matrices and singular value decompositions

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.28	−0.75	0.45	−0.20	0.12	−0.33
4	0.00	0.00	0.58	0.00	−0.58	0.58
5	−0.53	0.29	0.63	0.19	0.41	−0.22

- One column per document.
- V^T is an orthonormal matrix: (i) Row vectors have unit length. (ii) Any two distinct row vectors are orthogonal to each other.

- These are again the semantic dimensions that capture distinct topics like politics, sports and economics.
- Each number *vij* in the matrix indicates how strongly related document *i* is to the topic represented by semantic dimension *j*.

Outline

- 18.1 Linear algebra review
- 18.2 Term-document matrices and singular value decompositions
- 18.3 Low-rank approximations
- 18.4 Latent semantic indexing
- 18.5 References and further reading

18.3 Low-rank approximations

U	1	2	3	4	5
ship	−0.44	−0.30	0.00	0.00	0.00
boat	−0.13	−0.33	0.00	0.00	0.00
ocean	−0.48	−0.51	0.00	0.00	0.00
wood	−0.70	0.35	0.00	0.00	0.00
tree	−0.26	0.65	0.00	0.00	0.00

Σ_2	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00

$$C_2 = U\Sigma_2 V^T$$

C_2	d_1	d_2	d_3	d_4	d_5	d_6
ship	0.85	0.52	0.28	0.13	0.21	−0.08
boat	0.36	0.36	0.16	−0.20	−0.02	−0.18
ocean	1.01	0.72	0.36	−0.04	0.16	−0.21
wood	0.97	0.12	0.20	1.03	0.62	0.41
tree	0.12	−0.39	−0.08	0.90	0.41	0.49

C_2 as a two-dimensional representation of the matrix C

- Reducing the dimensionality to **2**

18.3 Low-rank approximations

- Why the reduced matrix C_2 is better than C?

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

C_2	d_1	d_2	d_3	d_4	d_5	d_6
ship	0.85	0.52	0.28	0.13	0.21	−0.08
boat	0.36	0.36	0.16	−0.20	−0.02	−0.18
ocean	1.01	0.72	0.36	−0.04	0.16	−0.21
wood	0.97	0.12	0.20	1.03	0.62	0.41
tree	0.12	−0.39	−0.08	0.90	0.41	0.49

Similarity of d_2 and d_3 in the original space: 0.
Similarity of d_2 and d_3 in the reduced space:
$0.52 * 0.28 + 0.36 * 0.16 + 0.72 * 0.36 + 0.12 * 0.20 + -0.39 * -0.08 \approx 0.52$

- “boat” and “ship” are semantically similar. The “reduced” similarity measure reflects this.

Outline

- 18.1 Linear algebra review
- 18.2 Term-document matrices and singular value decompositions
- 18.3 Low-rank approximations
- 18.4 Latent semantic indexing
- 18.5 References and further reading

18.4 Latent semantic indexing

- Using SVD for this purpose (in previous slides) is called latent semantic indexing or LSI.
- LSI addresses the problems of synonymy and semantic relatedness.
  - Standard vector space: Synonyms contribute nothing to document similarity.
  - Desired effect of LSI: Synonyms contribute strongly to document similarity (it will map synonyms to the same dimension).
- LSI usually increases **recall** and hurts **precision**.

18.4 Latent semantic indexing

- Implementation
  - Compute SVD of term-document matrix
  - Reduce the space and compute reduced document representations
  - Map the query into the reduced space $\boxed{\vec{q}_k = \Sigma_k^{-1} U_k^T \vec{q}}$
    - $C_k = U_k \Sigma_k V_k^T \Leftrightarrow \Sigma_k^{-1} U_k^T \underline{C_k} = V_k^T \Rightarrow \boxed{\Sigma_k^{-1} U_k^T \underline{C} = V_k^T}$
  - Compute similarity of q_k with all reduced documents in V_k.
  - Output ranked list of documents as usual

18.4 Latent semantic indexing

- Clustering

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

=

U	1	2	3	4	5
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
wood	−0.70	0.35	0.15	−0.58	0.16
tree	−0.26	0.65	−0.41	0.58	−0.09

×

Σ	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	1.28	0.00	0.00
4	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	0.00	0.39

×

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.28	−0.75	0.45	−0.20	0.12	−0.33
4	0.00	0.00	0.58	0.00	−0.58	0.58
5	−0.53	0.29	0.63	0.19	0.41	−0.22

- Each of the *k* dimensions of the reduced space is one cluster.
- If the value of the LSI representation of document *d* on dimension *k* is x, then x is the soft membership of *d* in topic k.
- This soft membership can be positive or negative.

Summary

- 18.1 Linear algebra review
- 18.2 Term-document matrices and singular value decompositions
- 18.3 Low-rank approximations
- 18.4 Latent semantic indexing
- 18.5 References and further reading