Latent Semantic Indexing

Debapriyo Majumdar
Indian Statistical Institute
debapriyo@isical.ac.in

The problem of synonymy and polysemy

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	q
amazon	0.6	0.9	0.7					
order	0.5	0.8						**1**
dispatch	1.0							
ship		0.2		0.5	0.3	0.2		**1**
ocean				0.8			0.1	
sea					0.4	0.7	0.5	
captain				0.9		0.8	0.3	
jungle			0.9					
anaconda			0.4					
cosine	0.23	0.51	0	0.27	0.42	0.13	0	
Rank	4	1	6	3	2	5	7	

- Commonly occurring in human languages
- Synonymy: two different words with (almost) the same meanings
  - Examples: (Ocean, Sea), (Dispatch, Ship)
  - Often, synonyms do not co-occur in the same document
- Polysemy: same word with different meanings
  - Examples: amazon (company or jungle), order (purchase, or authoritative instruction)
- Problem: ranking in vector space model suffers

Singular Value Decomposition

- If A is an $m \times n$ matrix with rank r, then there exists a factorization of A as

$$\underset{m \times n}{A} = \underset{m \times r}{U} \quad \underset{r \times r}{\Sigma} \quad \underset{r \times n}{V^T}$$

where U $(m \times r)$ and V $(n \times r)$ are orthogonal matrices, and Σ $(r \times r)$ is a diagonal matrix.

$\Sigma = (\sigma_{ij})$, where $\sigma_{ii} = \sigma_i$, for $i = 1,\ldots,r$ are the singular values of A, with $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_r \geq 0$.

- Columns of U (V) are the left (right) singular vectors of A.

Connection with Eigen Decomposition

- Consider $C = AA^T$ (a symmetric matrix)
- We have the SVD of A as $A = U\Sigma V^T$.
- So, we have

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

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

Since V is an orthogonal matrix, $V^T V = I$ (identity matrix).

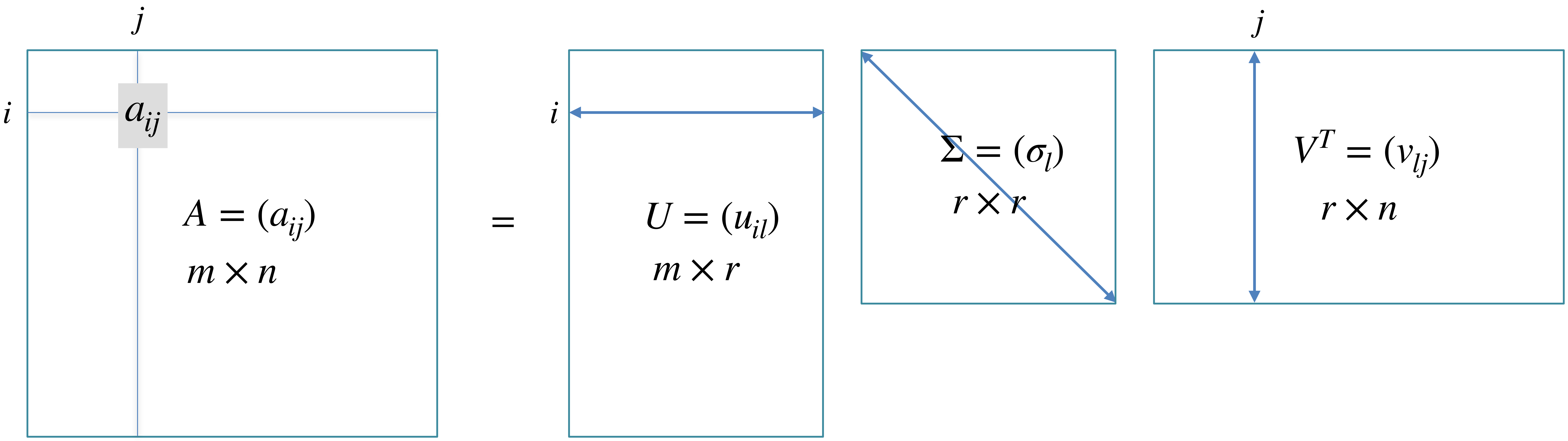
$$= U\Sigma\Sigma^T U^T$$

Σ is diagonal, so $\Sigma^T = \Sigma$.

$$= U\Sigma^2 U^T$$

- This is the Eigen-decomposition of C.
  - The columns of U are the eigenvectors.
  - The (diagonal) entries of Σ^2 are the eigenvalues.
    - The eigenvalues are simply $\sigma_1^2, \sigma_2^2, \ldots, \sigma_r^2$.

Singular vectors as principal components



$$a_{ij} = u_{i,1}\sigma_1 v_{1,j} \quad + \quad \cdots \quad + \quad u_{i,k}\sigma_k v_{k,j} \quad + \quad \cdots \quad + \quad u_{i,r}\sigma_r v_{rj}$$

Along first singular vectors, scaled by the first singular value

Along the k-th singular vectors, scaled by the k-th singular value

Along the r-th singular vectors, scaled by the r-th singular value

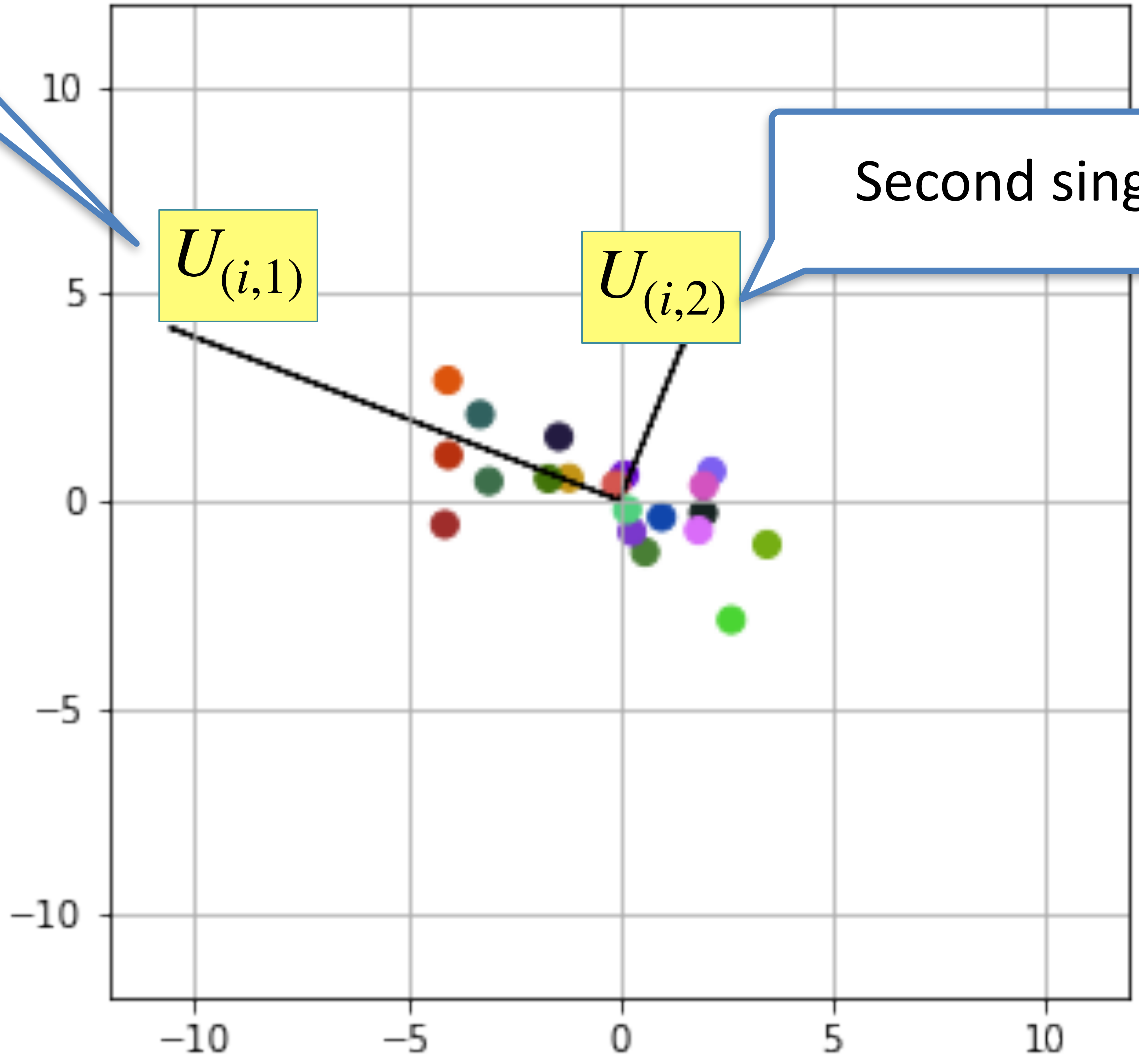
Singular vectors as principal components

$$a_{ij} = u_{i,1}\sigma_1 v_{1,j} \quad + \quad \cdots \quad + \quad u_{i,k}\sigma_k v_{k,j} \quad + \quad \cdots \quad + \quad u_{i,r}\sigma_r v_{rj}$$

First singular vector (principal component)

Fact: the first singular vector is the direction along which the data has maximum variance (intuitively, retains most separation).

Fact: If the projection along the first k singular vectors is removed, then the $k+1$-st vector is the direction along which the variance is maximized.



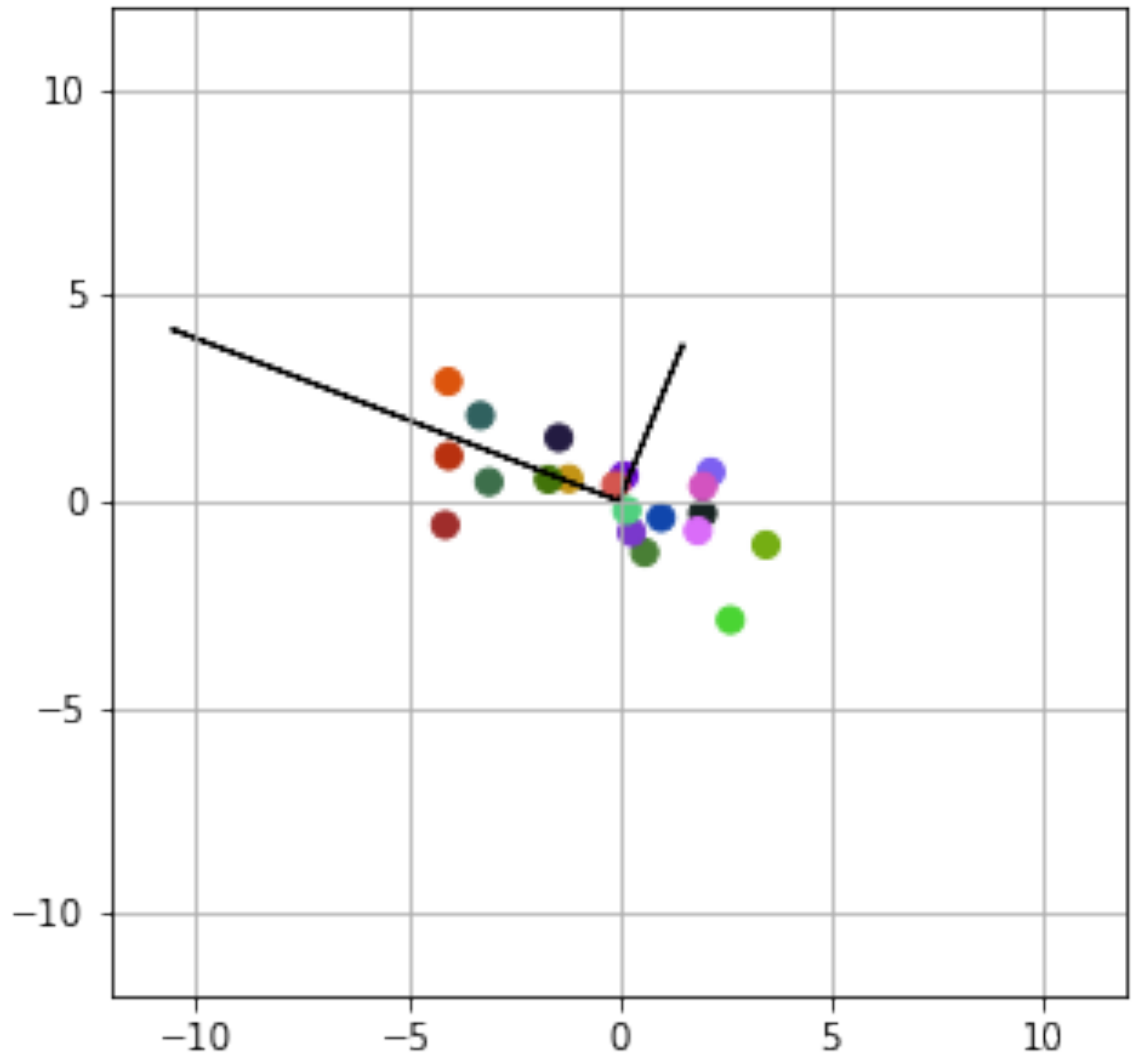
Second singular

Idea: consider the first "few" singular vectors. The most important information will be retained along those

Idea: discard the rest of the singular vectors, along those directions we have the noise which we better get rid of.

Note: in this 2-D example, there are only 2 singular vectors, so the first one is important and the other one is the noise.

The idea of latent concepts

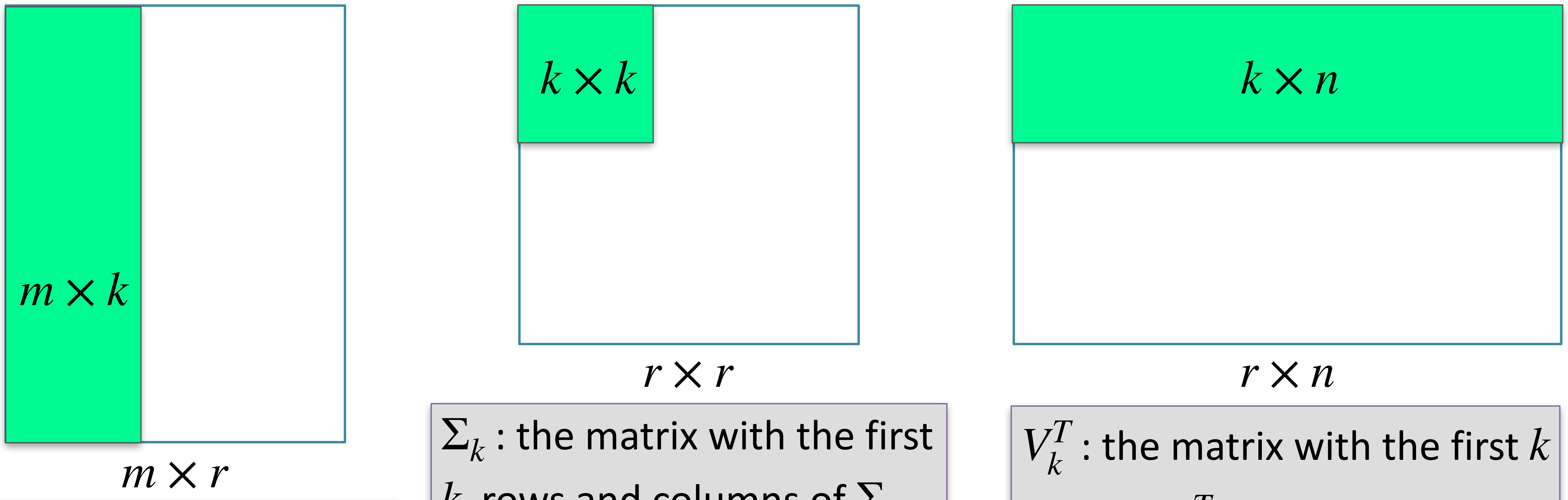


- An $m \times n$ term document matrix in the vector-space model
- Each document is a vector in the m dimensional term-space
  - Each term is a "feature" (orthogonal to each other)
  - But we know in reality not all terms are actually so
- The assumption of concept-space: there are k underlying (latent) primary concepts defining the semantics of the data
  - Each such concept may be a combination of some terms
  - The number of concepts << number of distinct terms
- Aim: the singular vectors (principal components) represent the concepts

Keep the information along the first k singular vectors and discard the rest (noise)

LSI: Low rank approximation

$$a_{ij} = u_{i,1}\sigma_1 v_{1,j} \quad + \quad \cdots \quad + \quad u_{i,k}\sigma_k v_{k,j} \quad \cancel{+ \quad \cdots \quad + \quad u_{i,r}\sigma_r v_{rj}}$$



U_k : the matrix with the first k columns of U (first k singular vectors)

Σ_k : the matrix with the first k rows and columns of Σ (first k singular values)

V_k^T : the matrix with the first k rows of V^T (first k right singular vectors)

Low rank approximation of A

$$A_k = U_k \Sigma_k V_k^T$$

Contains the information along only the first k singular vectors (and values)

LSI: Dimension Reduction

- The low-rank approximation does not reduce dimension
- $A_k = U_k \Sigma_k V_k^T$ is $m \times n$
- The original term-document matrix A was very sparse, but A_k is dense
  - Computationally expensive, infeasible for any large m and n
- Dimension reduction: same idea, differently implemented
  - Transform the document vectors to the lower (k) dimensional space spanned by the first k singular vectors
  - Map document $d \mapsto U_k^T d$, query $q \mapsto U_k^T q$ from the term-space to the concept-space
  - Equivalently the term-document matrix $A \mapsto U_k^T A$, concept-document matrix ($k \times n$)
    - Documents are vectors in the concept space
  - Compute cosine similarity in the k dimensional concept space

$$\text{sim}_{\text{concept}}(q, d) = \frac{(U_k^T q)^T U_k^T d}{\|U_k^T q\| \|U_k^T d\|}$$

References

- Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze. "Introduction to Information Retrieval", Cambridge University Press. 2008.