Latent Semantic Models

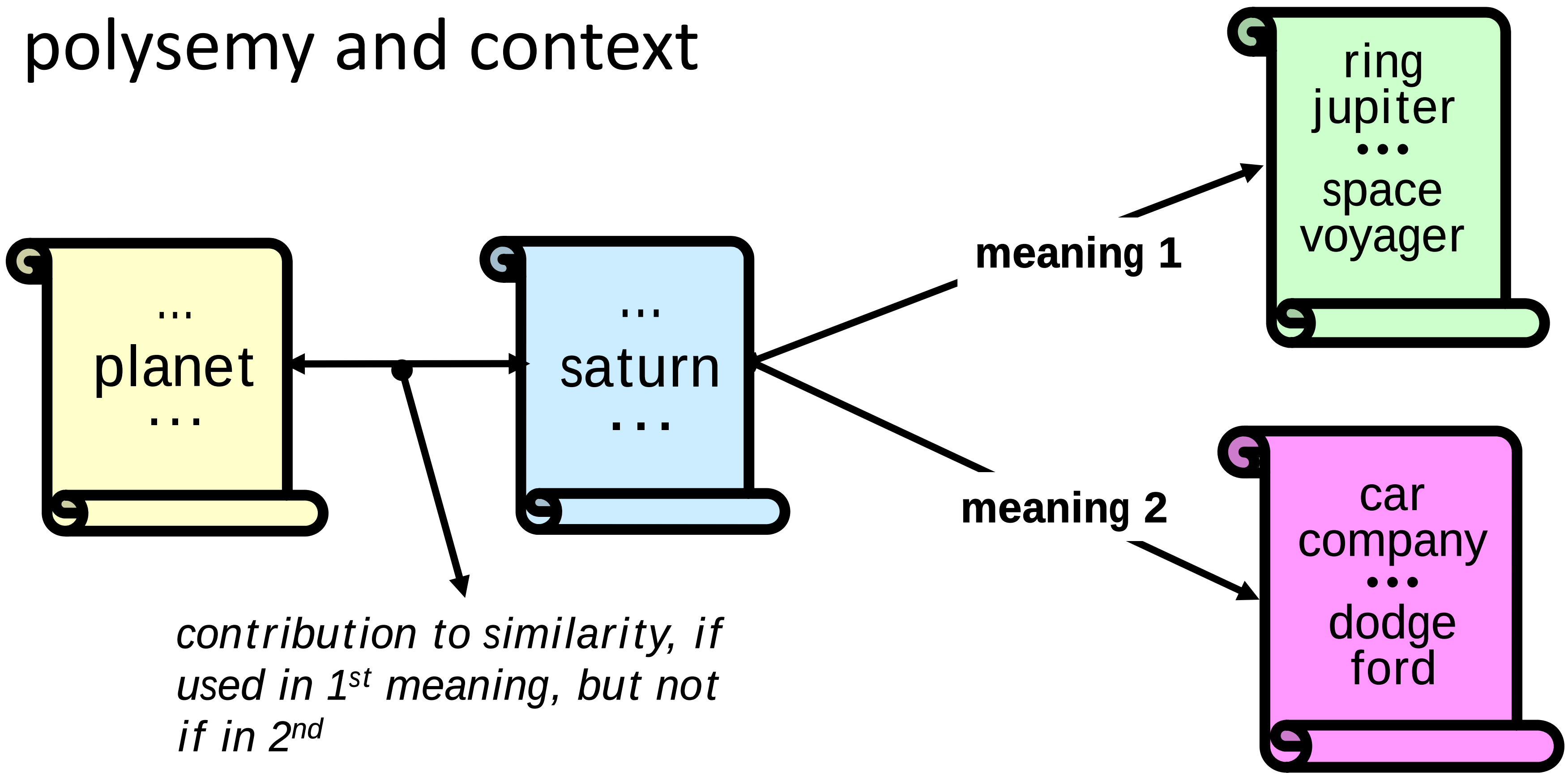
Reference: Introduction to Information Retrieval
by C. Manning, P. Raghavan, H. Schutze

Problems with Lexical Semantics

- Ambiguity and association in natural language
  - **Polysemy**: Words often have a **multitude of meanings** and different types of usage *(more severe in very heterogeneous collections).*
  - The basic IR models are unable to discriminate between different meanings of the same word.
  - **Synonymy**: Different terms may have an **identical or a similar meaning** (weaker: words indicating the same topic).
  - No associations between words are made in the vector space representation.

Polysemy and Context

- Document similarity on single word level: polysemy and context



Latent Semantic Indexing (LSI)

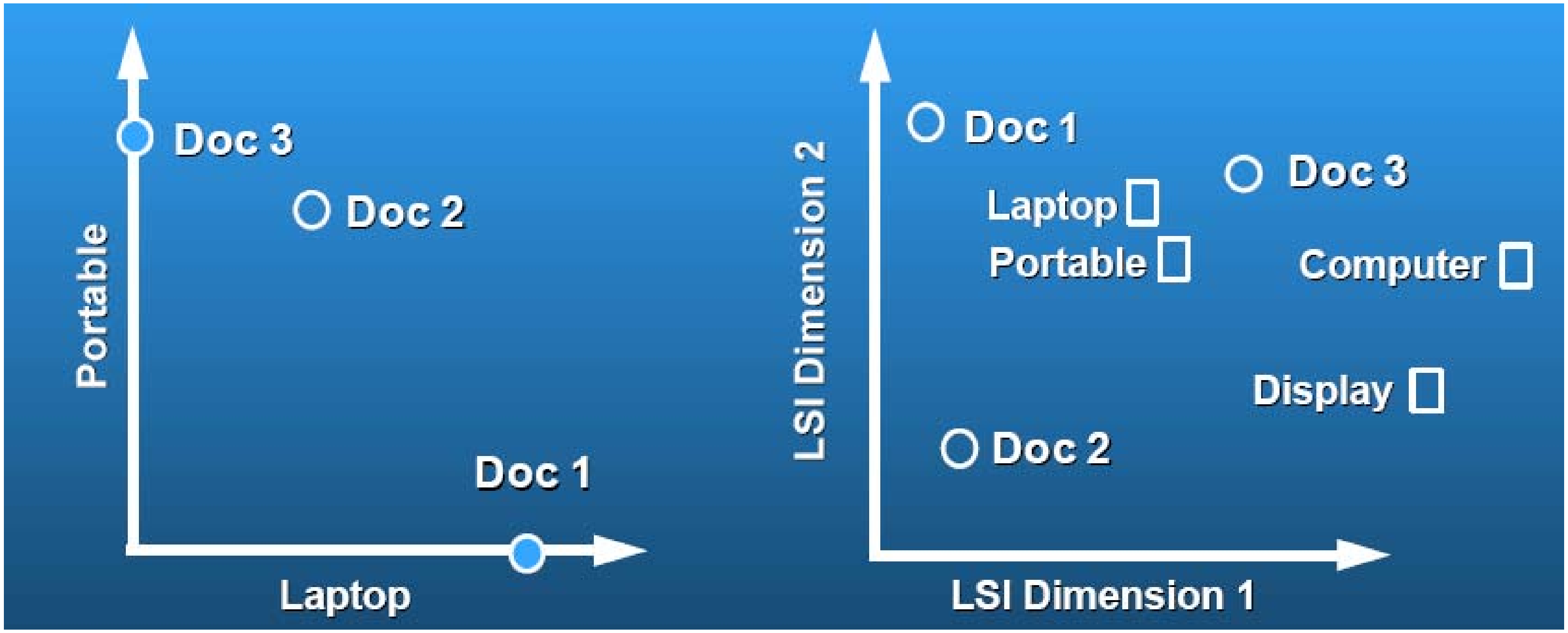
- Perform a **low-rank approximation** of **document-term matrix** (typical rank **100-300**)
- General idea
  - Map documents (*and* terms) to a **low-dimensional** representation.
  - Design a mapping such that the low-dimensional space reflects **semantic associations** (latent semantic space).
  - Compute document similarity based on the **inner product** in this **latent semantic space**

Goals of LSI

- Similar terms map to similar location in low dimensional space
- Noise reduction by dimension reduction

Latent Semantic Analysis

- **Latent semantic space**: illustrating example



courtesy of Susan Dumais

Latent Semantic Analysis

- Latent Semantic Analysis (LSA) is a particular application of Singular Value Decomposition (SVD) to a $M \times N$ term-document matrix A representing M words and their co-occurrence with *N* documents.
- SVD factorizes any such rectangular $M \times N$ matrix A into the product of three matrices U, Σ, and V^T.

Latent Semantic Analysis

- In the $M \times r$ matrix *U*, each of the *u* rows still represents a word.
- Each column now represents one of *r* dimensions in a latent space. Sometimes we call it "topic" or "concept".
- The *r* column vectors are orthogonal to each other.
- For two vectors such as v_1 and v_2, they are orthogonal if $v_1 \cdot v_2 = v_1^T v_2 = 0$

Latent Semantic Analysis

- The columns are ordered by the amount of variance in the original dataset each accounts for.
- The number of such dimensions *r* is the **rank** of X (the rank of a matrix is the number of linearly independent rows).

Latent Semantic Analysis

- Σ is a diagonal $r \times r$ matrix, with **singular values** along the diagonal, expressing the importance of each dimension.
- The $r \times N$ matrix V^T still represents documents, but each row now represents one of the new latent dimensions and the *r* row vectors are orthogonal to each other.

Latent Semantic Analysis

- By using only the first *k* dimensions, of U, Σ, and V instead of all r dimensions, the product of these 3 matrices becomes a least-squares approximation to the original A.
- Since the first dimensions encode the most variance, one way to view the reconstruction is thus as modeling the most important information in the original dataset.

Latent Semantic Analysis

- SVD applied to co-occurrence matrix A:

$$\underset{M \times N}{\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix}} = \underset{M \times r}{\begin{bmatrix} & & \\ & U & \\ & & \end{bmatrix}} \underset{r \times r}{\begin{bmatrix} \sigma_1 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & 0 \\ 0 & 0 & \sigma_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_r \end{bmatrix}} \underset{r \times N}{\begin{bmatrix} & & \\ & V^T & \\ & & \end{bmatrix}}$$

Latent Semantic Analysis

- Taking only the top k, $k \leq r$ dimensions after the SVD is applied to the co-occurrence matrix A:

$$\underset{M \times N}{\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix}} = \underset{M \times k}{\begin{bmatrix} & & \\ & U_k & \\ & & \end{bmatrix}} \underset{k \times k}{\begin{bmatrix} \sigma_1 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & 0 \\ 0 & 0 & \sigma_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_k \end{bmatrix}} \underset{k \times N}{\begin{bmatrix} & V^T & \end{bmatrix}}$$

SVD factorizes a matrix into a product of three matrices, U, Σ, and V^T. Taking the first k dimensions gives a $M \times k$ matrix U_k that has one k-dimensioned row per word

Related Linear Algebra Background

Eigenvalues & Eigenvectors

- **Eigenvectors** (for a square $m \times m$ matrix $\mathbf{S}$)

$$\mathbf{S}\mathbf{v} = \lambda \mathbf{v}$$

(right) eigenvector $\mathbf{v} \in \mathbb{R}^m \neq \mathbf{0}$

eigenvalue $\lambda \in \mathbb{R}$

Example

$$\begin{pmatrix} 6 & -2 \\ 4 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \end{pmatrix} = 2 \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

- How many eigenvalues are there at most?

$$\mathbf{S}\mathbf{v} = \lambda \mathbf{v} \iff (\mathbf{S} - \lambda \mathbf{I})\,\mathbf{v} = \mathbf{0}$$

only has a non-zero solution if $|\mathbf{S} - \lambda \mathbf{I}| = 0$

This is a mth order equation in λ which can have **at most m distinct solutions** (roots of the characteristic polynomial) – <u>can be complex even though **S** is real.</u>

Matrix-vector multiplication

$$S = \begin{bmatrix} 30 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

has eigenvalues 30, 20, 1 with corresponding eigenvectors

$$v_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad v_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad v_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

On each eigenvector, S acts as a multiple of the identity matrix: but as a (usually) different multiple on each.

Any vector (say $x = \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix}$) can be viewed as a combination of the eigenvectors: $x = 2v_1 + 4v_2 + 6v_3$

Matrix vector multiplication

- Thus a matrix-vector multiplication such as *Sx* (*S*, *x* as in the previous slide) can be rewritten in terms of the eigenvalues/vectors:

$$Sx = S(2v_1 + 4v_2 + 6v_3)$$

$$Sx = 2Sv_1 + 4Sv_2 + 6Sv_3 = 2\lambda_1 v_1 + 4\lambda_2 v_2 + 6\lambda_3 v_3$$

$$Sx = 60v_1 + 80v_2 + 6v_3$$

- Even though *x* is an arbitrary vector, the action of *S* on *x* is determined by the eigenvalues/vectors.

Matrix vector multiplication

- Suggestion: the effect of “small” eigenvalues is small.
- If we ignored the smallest eigenvalue (1), then instead of

$$\begin{pmatrix} 60 \\ 80 \\ 6 \end{pmatrix} \quad \text{we would get} \quad \begin{pmatrix} 60 \\ 80 \\ 0 \end{pmatrix}$$

- These vectors are similar (in cosine similarity, etc.)

Left Eigenvectors

- In a similar fashion, the left eigenvectors of a square matrix *C* are *y* such that :

$$y^T C = \lambda y^T$$

where λ is the corresponding eigenvalue:

- Consider a square matrix S with eigenvector v. We have: $Sv = \lambda v$

$$v^T S^T = \lambda v^T$$

Recall that
$(AB)^T = B^T A^T$

- Therefore, the eigenvalue of the right eigenvector is the same as the eigenvalue of the left eigenvector of the transposed matrix.

Eigenvalues & Eigenvectors

$$Sv_{\{1,2\}} = \lambda_{\{1,2\}} v_{\{1,2\}}$$

For a symmetric matrix *S*, eigenvectors for distinct eigenvalues are **orthogonal**

$$\text{For } \lambda_1 \neq \lambda_2,\ v_1 \bullet v_2 = v_1^T v_2 = 0$$

Eigenvalues & Eigenvectors

All eigenvalues of a real symmetric matrix are **real**.

for complex λ, if $|S - \lambda I| = 0$ and $\mathrm{S} = \mathrm{S}^{\mathrm{T}} \Rightarrow \lambda \in \Re$

All eigenvalues of a positive semidefinite matrix are **non-negative**

$$\forall w \in \Re^n, w^T S w \geq 0, \text{ then if } \mathrm{Sv} = \lambda \mathrm{v} \Rightarrow \lambda \geq 0$$

For any matrix *A*, $A^T A$ is positive semidefinite

Example

- Let

$$S = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

Real, symmetric.

- Then

$$S - \lambda I = \begin{bmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{bmatrix} \Rightarrow$$

$$|S - \lambda I| = (2-\lambda)^2 - 1 = 0.$$

- The eigenvalues are 1 and 3 (nonnegative, real).
- The eigenvectors are orthogonal (and real):

$$\begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Plug in these values and solve for eigenvectors.

Eigen/diagonal Decomposition

- Let $\mathbf{S} \in \mathbb{R}^{m \times m}$ be a **square** matrix with ***m*** **linearly independent eigenvectors** (a "non-defective" matrix)

- **Theorem**: Exists an **eigen decomposition**

  diagonal

$$\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$$

Unique for distinct eigen-values

  - (cf. matrix diagonalization theorem)

- Columns of ***U*** are **eigenvectors** of ***S***

- Diagonal elements of $\mathbf{\Lambda}$ are **eigenvalues** of $\mathbf{S}$

$$\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_m), \quad \lambda_i \geq \lambda_{i+1}$$

Diagonal decomposition: why/how

Let ***U*** have the eigenvectors as columns: $U = \begin{bmatrix} v_1 & \dots & v_n \end{bmatrix}$

Then, ***SU*** can be written

$$SU = S\begin{bmatrix} v_1 & \dots & v_n \end{bmatrix} = \begin{bmatrix} \lambda_1 v_1 & \dots & \lambda_n v_n \end{bmatrix} = \begin{bmatrix} v_1 & \dots & v_n \end{bmatrix}\begin{bmatrix} \lambda_1 & & \\ & \dots & \\ & & \lambda_n \end{bmatrix}$$

Thus $SU=U\Lambda$, or $U^{-1}SU=\Lambda$

And $S=U\Lambda U^{-1}$.

Diagonal decomposition - example

Recall $S = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}; \lambda_1 = 1, \lambda_2 = 3.$

The eigenvectors $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$ and $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ form $U = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$

Inverting, we have $U^{-1} = \begin{bmatrix} 1/2 & -1/2 \\ 1/2 & 1/2 \end{bmatrix}$

Recall $UU^{-1} = 1$.

Then, $\boldsymbol{S=U\Lambda U^{-1}} =$ $\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 1/2 & -1/2 \\ 1/2 & 1/2 \end{bmatrix}$

Example continued

Let's divide ***U*** (and multiply U^{-1}) by $\sqrt{2}$

Then, ***S*=** $\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$

Q Λ $(Q^{-1} = Q^T)$

Symmetric Eigen Decomposition

- If $\mathbf{S} \in \mathbb{R}^{m \times m}$ square **symmetric** matrix with *m* linearly independent eigenvectors:
- **Theorem**: There exists a (unique) **eigen decomposition**

$$S = Q \Lambda Q^T$$

- where ***Q*** is **orthogonal:**
  - $Q^{-1} = Q^T$
  - Each column v_i of ***Q*** are normalized eigenvectors
  - Columns are orthogonal (also called orthonormal basis)

$$v_i \bullet v_j = v_i^T v_j = 0 \quad \text{if} \quad i \neq j$$

$$v_i \bullet v_i = v_i^T v_i = 1$$

Connection to Singular Value Decomposition (SVD)

- Recall a $M \times N$ term-document matrix A representing M words and their co-occurrence with N documents.
- By multiplying A by its transposed version,

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

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

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

- Note that the left-hand side is a squared symmetric matrix, and the right-hand side represents its symmetric diagonal decomposition.
- SVD factorizes any such rectangular $M \times N$ matrix A into the product of three matrices U, Σ, and V^T.

Singular Value Decomposition (SVD)

Singular Value Decomposition

For an $M \times N$ matrix $\mathbf{A}$ of rank r there exists a factorization (Singular Value Decomposition = **SVD**) as follows:

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

U is $M\times M$; Σ is $M\times N$; V is $N\times N$

The columns of ***U*** are normalized orthogonal eigenvectors of AA^T.

The columns of ***V*** are normalized orthogonal eigenvectors of A^TA.

Eigenvalues $\lambda_1 \ldots \lambda_r$ of AA^T are the eigenvalues of A^TA.

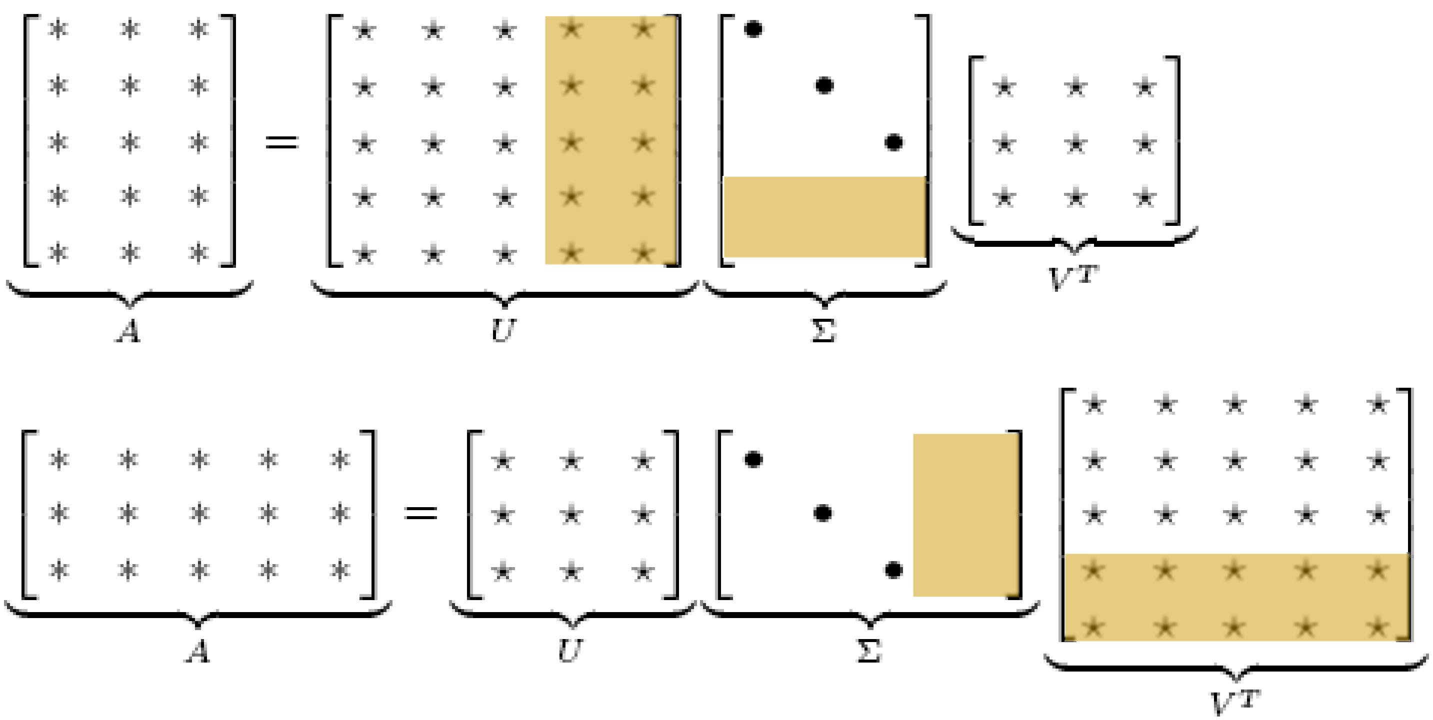
$$\sigma_i = \sqrt{\lambda_i} \qquad \Sigma = diag(\sigma_1 \ldots \sigma_r)$$

Singular values.

Recall that the rank of a matrix is the maximum number of linearly independent rows or columns

Singular Value Decomposition

- Illustration of SVD dimensions and sparseness



SVD example

Let $A = \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$

Thus M=3, N=2. Its SVD is

$$\begin{bmatrix} 2/\sqrt{6} & 0 & 1/\sqrt{3} \\ -1/\sqrt{6} & 1/\sqrt{2} & 1/\sqrt{3} \\ 1/\sqrt{6} & 1/\sqrt{2} & -1/\sqrt{3} \end{bmatrix} \begin{bmatrix} \sqrt{3} & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

Typically, the singular values arranged in decreasing order.

Low-rank Approximation

- SVD can be used to compute optimal **low-rank approximations**.
- Approximation problem: Find ***X*** such that

$$\min_{X: rank(X)=k} \|A - X\|_F \leftarrow \text{Frobenius norm}$$

$$\|A\|_F \equiv \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}.$$

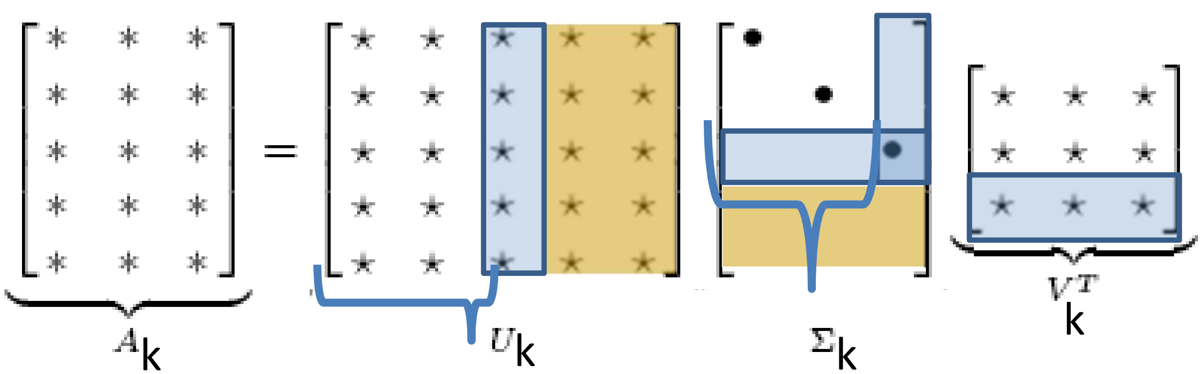
- Let the solution be denoted by $\boldsymbol{A_k}$ (rank $\boldsymbol{k}$)
- A_k is the best approximation of A.
- Typically, we want $k << r$.

Low-rank Approximation

- Solution via SVD

$$A_k = U \operatorname{diag}(\sigma_1, \ldots, \sigma_k, \underbrace{0, \ldots, 0}) V^T$$

set smallest r-k
singular values to zero



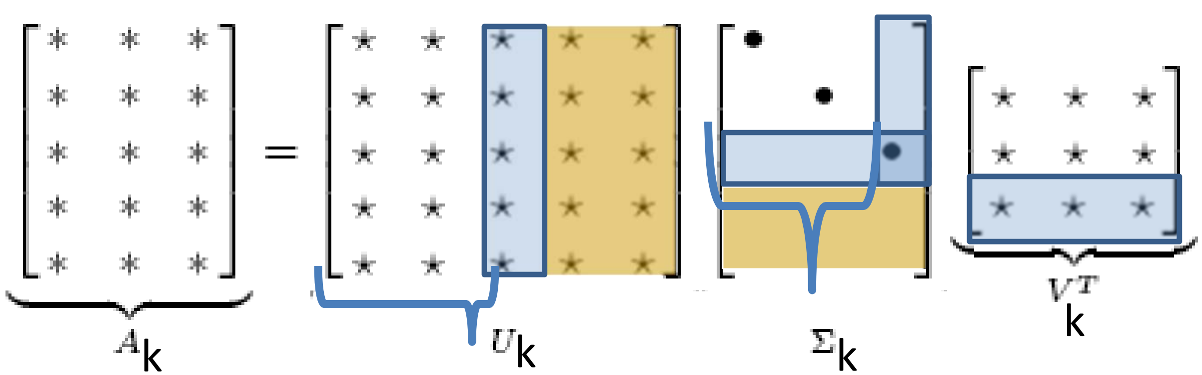
$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

column notation: sum of rank 1 matrices

Reduced SVD

- If we retain only *k* singular values, and set the rest to 0, we don't need the matrix parts in red
- Then Σ_k is $k \times k$, U_k is $M \times k$, V_k^T is $k \times N$, and A_k is $M \times N$

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



- This is referred to as the reduced SVD

Approximation error

- How good (bad) is this approximation?
- It's the best possible, measured by the Frobenius norm of the error:

$$\min_{X:rank(X)=k} \left\| A - X \right\|_F = \left\| A - A_k \right\|_F = \sigma_{k+1}$$

where the σ_i are ordered such that $\sigma_i \geq \sigma_{i+1}$.
Suggests why Frobenius error drops as *k* increased.

SVD Low-rank approximation

- Suppose that the term-doc matrix A may have M=50000, N=10 million (and rank close to 50000)
- We can construct an approximation A_{100} with rank 100.
  - Of all rank 100 matrices, it would have the lowest Frobenius error.

Latent Semantic Indexing via the SVD

What it is

- From term-doc matrix A, we compute the approximation A_{k}.
- There is a row for each term and a column for each doc in A_k
- Thus docs live in a space of $k \ll r$ dimensions
  - These dimensions are not the original axes

Performing the maps

- Each row and column of *A* gets mapped into the *k*-dimensional LSI space, by the SVD.

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

The columns of U_k are normalized

  - As a result: $$V_k = A_k^T U_k \Sigma_k^{-1}$$

- A query *q* is also mapped into this space, by

$$q_k = q^T U_k \Sigma_k^{-1}$$

Query NOT a sparse vector

Performing the maps

- Conduct similarity calculation under the low dimensional space (k)
- Claim – this is not only the mapping with the best (Frobenius error) approximation to A, but also *improves* retrieval.

Empirical evidence

- Experiments on TREC 1/2/3 – Dumais
- Lanczos SVD code (available on netlib) due to Berry used in these experiments
  - Running times quite long
- Dimensions – various values 250-350 reported.

Empirical evidence

- Precision at or above median TREC precision
  - Top scorer on almost 20% of TREC topics
- Slightly better on average than straight vector spaces
- Effect of dimensionality:

Dimensions	Precision
250	0.367
300	0.371
346	0.374

Failure modes

- Negated phrases
  - TREC topics sometimes negate certain query/terms phrases – precludes automatic conversion of topics to latent semantic space.
- Boolean queries
  - As usual, free-text/vector space syntax of LSI queries precludes (say) “Find any doc having to do with the following 5 companies”