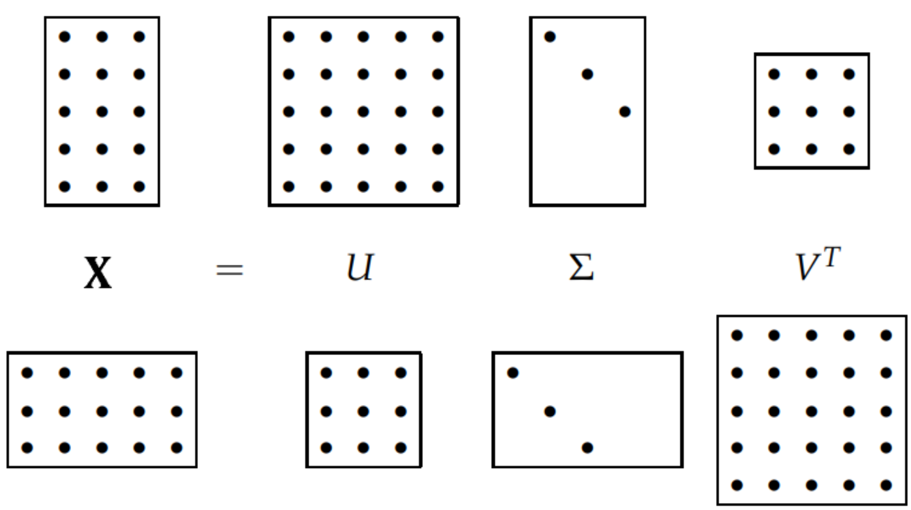
# SVD, SVD applications to LSA, non-negative matrix factorizations

Presented By: Sumedha Singla

## Singular Value Decomposition (SVD)



# Singular Value Decomposition (SVD)

SVD of a matrix $\mathbf{X}$

$$\mathbf{X}_{n \times d} = \mathbf{U}_{n \times n}\mathbf{\Sigma}_{n \times d}\mathbf{V}_{d \times d}^{T} \quad \text{or}$$
$$\mathbf{X}_{n \times d} = \mathbf{U}_{n \times k}\mathbf{\Sigma}_{k \times k}\mathbf{V}_{k \times d}^{T}$$

- $\mathbf{X}$: A set of n points in $\mathbb{R}^{d}$ with rank k
- $\mathbf{U}$ : Left Singular Vectors of $\mathbf{X}$
- $\mathbf{V}$ : Right Singular Vectors of $\mathbf{X}$
- $\mathbf{\Sigma}$: Rectangular diagonal matrix with positive real entries.

$$\mathbf{X} = \begin{bmatrix} u_1 & \dots & u_k & \dots & u_n \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{bmatrix} \begin{bmatrix} v_1^T & \dots & v_k^T & \dots & v_d^T \end{bmatrix}$$

$$\mathbf{X} = \mathbf{U\Sigma V}^{T} = u_1\sigma_1 v_1^T + \dots + u_k\sigma_k v_k^T = \sum_{i=1}^{k} u_i\sigma_i v_i^T$$

# Singular Value Decomposition (SVD)

SVD of a matrix $\mathbf{X}$

$$\mathbf{X}\,\mathbf{v_i} = \sigma_i\,\mathbf{u_i}$$

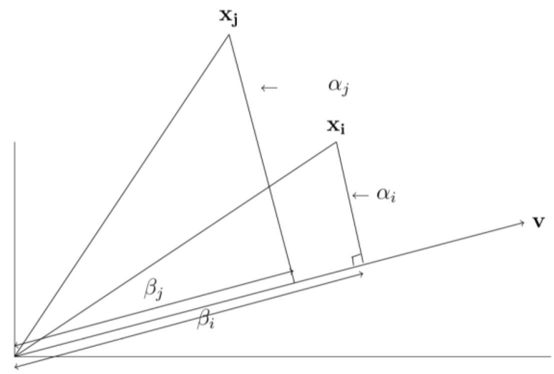
- Finding an orthogonal basis for the row space that gets transformed into an orthogonal basis for the column space.
- The columns of $\mathbf{U}$ and $\mathbf{V}$ are bases for the row and column spaces, respectively.
- $\mathbf{U}$ and $\mathbf{V}$ are orthonormal square matrix i.e

$$\mathbf{VV}^{T} = \mathbf{V}^{T}\mathbf{V} = \mathbf{I}$$
$$\mathbf{UU}^{T} = \mathbf{U}^{T}\mathbf{U} = \mathbf{I}$$

- Usually, $\mathbf{U} \neq \mathbf{V}$.

# Motivation

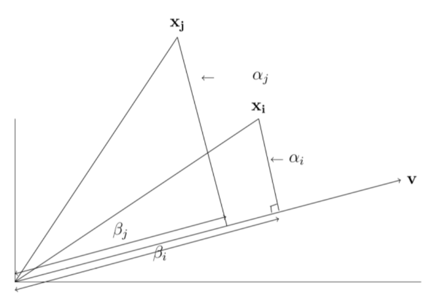
- Goal: Find the best k-dimensional subspace w.r.t $\mathbf{X}$ (Project $\mathbf{X}$ to $\mathbb{R}^k$ where $k < d$)
  - minimize the sum of the squares of the perpendicular distances of the points to the subspace
- Consider a set of 2d points. $\mathbf{X}_{n \times 2}$, $\mathbf{x}_i \in \mathbb{R}^2; 1 \leq i \leq n$
  - Goal: Find the best fitting line through origin w.r.t $\mathbf{X}$
  - Here, $k = 1$
  - **Best least square fit**
    - Minimize $\sum \alpha_i^2$ or
    - Maximize $\sum \beta_i^2$ i.e projection of $\mathbf{x}_i$ on subspace




# Singular Vectors

$$\mathbf{X}\,\mathbf{v_i} = \sigma_i\,\mathbf{u_i}$$

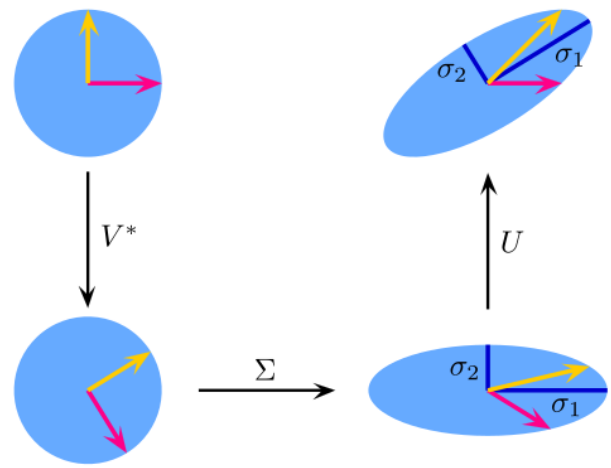
- $\mathbf{v}$: A unit vector in the direction of the best fitting line through origin w.r.t $\mathbf{X}$
- $\beta_i = |x_i \,.\, \mathbf{v}|$
- **Best least square fit**
  - Maximizing $\sum \beta_i^2 = |\mathbf{X}\,.\,\mathbf{v}|^2$
- First singular vector
  - $\mathbf{v}_1 = \arg\max_{|\mathbf{v}|=1} |\mathbf{X}\,.\,\mathbf{v}|$
- First singular value
  - $\sigma_1 = |\mathbf{X}\,.\,\mathbf{v}_1|$
- Greedy approach for subsequent singular vectors
  - Best fit line perpendicular to $\mathbf{v}_1$
  - $\mathbf{v}_2 = \arg\max_{\mathbf{v}\perp\mathbf{v}_1, |\mathbf{v}|=1} |\mathbf{X}\,.\,\mathbf{v}|$



$$\mathbf{X = U\,\Sigma\,V^T}$$

# Intuitive Interpretation

A composition of three geometrical transformations:
a rotation or reflection, a scaling,
and another rotation or reflection.




# Intuitive Interpretation

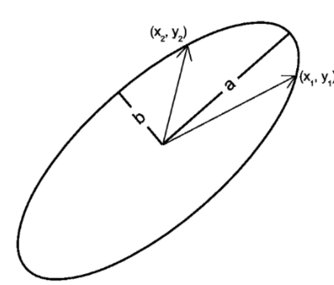
$$\mathbf{X = U\,\Sigma\,V^T}$$

- Consider a unit circle

$$\mathbf{x'.x' = 1}$$

- An ellipse of any size and orientation by stretching and rotating it.
- Consider 2-d points and fit an ellipse with major axis (a) and minor axes (b) to them.
- Consider,

$$\mathbf{S} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}, \mathbf{R} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$




- Any point can be transformed as

$$\mathbf{x' = x\,R\,S^{-1}}$$

- The equation of unit circle

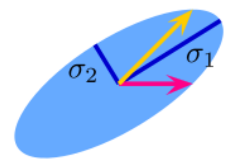
$$\mathbf{S^{-1}R^{T}x\,.\,x\,R\,S^{-1} = 1}$$

# Intuitive Interpretation

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

- Resulting matrix equation

$$\mathbf{S}^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{R}\mathbf{S}^{-1} = \mathbf{1}$$




- If we regard $\mathbf{X}$ as a collection of points, then
  - The singular values are the axes of a least squares fitted ellipsoid
  - $\mathbf{V}$ is orientation of the ellipsoid.
  - The matrix $\mathbf{U}$ is the projection of each of the points in $\mathbf{X}$ onto the axes.

# SVD Example

$\mathbf{X}_{n\times d} = \mathbf{U}_{n\times k}\boldsymbol{\Sigma}_{k\times k}\mathbf{V}^{\mathrm{T}}_{k\times d}$

- Natural Language Processing
- Documents with 2 concepts:
  - Computer Science (CS)
  - Medical Documents (MD)

Columns: data, inf., retrieval, brain, lung; Rows: CS (first four), MD (last three)

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

Term-Document Matrix
Row: 1 Document
Columns: 1 Term

Document-Concept
Similarity Matrix
Row: 1 Document
Columns: 1 Concept

Concept
Strength Matrix
Row: 1 Concept

Term-Concept Matrix
Row: 1 Concept
Column: 1 Term

## Eigen value decomposition

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

Eigen Vector

- An eigenvector of a square matrix $\mathbf{X}$ is a nonzero vector $\mathbf{v}$ such that multiplication by $\mathbf{X}$ alters only the scale of $\mathbf{v}$

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

  - $\lambda$: Eigen value
  - $\mathbf{v}$: Unit Eigen vector

Eigen Value Decomposition

$$\mathbf{X} = \mathbf{V}\,\mathrm{diag}(\boldsymbol{\lambda})\mathbf{V}^{-1} \text{ where}$$

  - Eigen vector matrix $\mathbf{V} = [\mathrm{v}_1, \ldots, \mathrm{v}_\mathrm{n}]$
  - Diagonal matrix $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_\mathrm{n}]$

More general form

$$\mathbf{X} = \mathbf{Q}\,\boldsymbol{\Lambda}\,\boldsymbol{Q}^{T}$$

## When is singular values same as eigen values

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

- Eigen value decomposition: $\mathbf{X} = \mathbf{Q}\,\boldsymbol{\Lambda}\,\boldsymbol{Q}^{T}$
- $\mathbf{X}$ needs
  - orthonormal eigen vectors to allow $\mathbf{U} = \mathbf{V} = \mathbf{Q}$.
  - Eigenvalues $\lambda \geq 0$ if $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}$.
- Hence, $\mathbf{X}$ must be a positive semi-definte (or definite) symmetric matrix.
- Eigen value decomposition is a special case of SVD.

# Calculating SVD using Eigen value decomposition

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

Rather than solving for U, V and Σ simultaneously, we multiply both sides by

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

$$\begin{aligned}\mathbf{X}^{\mathrm{T}}\mathbf{X} &= (\mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^{\mathrm{T}})^{\mathrm{T}}(\mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^{\mathrm{T}}) \\ &= \boldsymbol{V}\,\boldsymbol{\Sigma}^{\mathrm{T}}\,\boldsymbol{U}^{\mathrm{T}}\mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^{\mathrm{T}} \\ &= \boldsymbol{V}\,\boldsymbol{\Sigma}^{\mathrm{T}}\,\mathbf{\Sigma}\,\mathbf{V}^{\mathrm{T}} \\ &= \boldsymbol{V}\,\boldsymbol{\Sigma}^{2}\,\mathbf{V}^{\mathrm{T}}\end{aligned}$$

This is the form of eigen value decomposition. $\mathbf{X} = \mathbf{Q}\,\boldsymbol{\Lambda}\,\boldsymbol{Q}^{T}$

$\mathbf{V}$: The eigen vectors of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$.

$\mathbf{\Sigma}^{\mathrm{T}}\,\mathbf{\Sigma}$: The eigen value matrix of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$.

$\sigma_i = \sqrt{\lambda_{\mathrm{i}}}$

$\mathbf{U}$: The eigen vectors of $\mathbf{X}\mathbf{X}^{\mathrm{T}}$.

# SVD and Eigen value decomposition

$$\mathbf{X}\,\mathbf{v_i} = \sigma_{\mathrm{i}}\,\mathbf{u_i}$$

We know that,

$$\mathbf{u}_i^T\,\mathbf{u}_{\mathrm{j}} = \left(\frac{\mathbf{X}\,\mathbf{v}_{\mathrm{i}}}{\sigma_{\mathrm{i}}}\right)^{\mathrm{T}}\left(\frac{\mathbf{X}\,\mathbf{v}_{\mathrm{j}}}{\sigma_{\mathrm{j}}}\right)$$

$$\mathbf{u}_i^T\,\mathbf{u}_{\mathrm{j}} = \frac{\mathbf{v}_i^T\,\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{v}_{\mathrm{j}}}{\sigma_{\mathrm{i}}\,\sigma_{\mathrm{j}}} = \frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}}{\sigma_{\mathrm{i}}\,\sigma_{\mathrm{j}}}\,\mathbf{v}_i^T\,\mathbf{v}_{\mathrm{j}} = 0$$

$\mathbf{U}$: The orthonormal eigen vectors of $\mathbf{X}\,\mathbf{X}^{\mathrm{T}}$.

We can thus write,

$$\mathbf{X}\,\mathbf{X}^{\mathrm{T}}\mathbf{U} = \mathbf{U}\,\mathbf{\Sigma}^{2}$$

# Example SVD

- Consider $\mathbf{X} = \begin{bmatrix} 4 & 4 \\ -3 & 3 \end{bmatrix}$
- Compute $\mathbf{X}^{\mathrm{T}}\mathbf{X} = \begin{bmatrix} 4 & -3 \\ 4 & 3 \end{bmatrix}\begin{bmatrix} 4 & 4 \\ -3 & 3 \end{bmatrix} = \begin{bmatrix} 25 & 7 \\ 7 & 25 \end{bmatrix}$
- Orthogonal Eigen vector of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$
  - $\mathbf{v}_1 = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$ and $\mathbf{v}_2 = \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}$
- Eigen values of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$
  - ${\sigma_1}^2 = 32$ and ${\sigma_2}^2 = 18$
- We have,

$$\begin{bmatrix} 4 & 4 \\ -3 & 3 \end{bmatrix} = \begin{bmatrix} & \\ & \end{bmatrix}\begin{bmatrix} 4\sqrt{2} & 0 \\ 0 & 3\sqrt{2} \end{bmatrix}\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

# Example SVD

- Consider $\mathbf{X} = \begin{bmatrix} 4 & 4 \\ -3 & 3 \end{bmatrix}$
- Compute $\mathbf{X}\,\mathbf{X}^{\mathrm{T}} = \begin{bmatrix} 4 & 4 \\ -3 & 3 \end{bmatrix}\begin{bmatrix} 4 & -3 \\ 4 & 3 \end{bmatrix} = \begin{bmatrix} 32 & 0 \\ 0 & 18 \end{bmatrix}$
- Orthogonal Eigen vector of $\mathbf{X}\,\mathbf{X}^{\mathrm{T}}$
  - $\mathbf{u}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\mathbf{u}_2 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$
- We have,

$$\begin{bmatrix} 4 & 4 \\ -3 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} 4\sqrt{2} & 0 \\ 0 & 3\sqrt{2} \end{bmatrix}\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

## SVD vs Eigen Decomposition

- An eigen-decomposition is valid only for square matrix. Any matrix (even rectangular) has an SVD.
- In eigen-decomposition $\mathbf{X} = \mathbf{Q}\,\boldsymbol{\Lambda}\,\boldsymbol{Q}^T$, the eigen-basis ($\mathbf{Q}$) is not always orthogonal. The basis of singular vectors is always orthogonal.
- In SVD we have two singular-spaces (right and left).
- Computing the SVD of a matrix is more numerically stable.

## SVD and PCA

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

- The covariance matrix of $\mathbf{X}$ is given by

$$\mathbf{Cov} = \mathbf{X}^{\mathrm{T}}\,\mathbf{X}/(\mathbf{n}-\mathbf{1})$$

- The eigen value decomposition of $\mathbf{Cov}$ matrix

$$\mathbf{Cov} = \mathbf{Q}\,\boldsymbol{\Lambda}\,\boldsymbol{Q}^T$$

Where,

$\mathbf{Q}$ is a matrix of eigenvectors of $\mathbf{Cov}$ or principal axes of $\mathbf{X}$

$\boldsymbol{\Lambda}$ is a diagonal matrix with eigenvalues $\lambda_{\mathrm{i}}$ in the decreasing order on the diagonal.

## SVD and PCA

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

- We can rewrite covariance matrix of $\mathbf{X}$ as

$$\mathbf{Cov} = \mathbf{X}^{\mathrm{T}}\,\mathbf{X}/(\mathbf{n}-\mathbf{1})$$
$$\mathbf{Cov} = \mathbf{V}\,\mathbf{\Sigma}\,\mathbf{U}^{\mathrm{T}}\mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^{\mathrm{T}}/(\mathbf{n}-\mathbf{1})$$
$$= \mathbf{V}\frac{\mathbf{\Sigma}^2}{(n-1)}\,\mathbf{V}^{\mathrm{T}}$$

- Right singular vector $\mathbf{V}$ is the principal axes
- $\lambda_i = \sigma_i^2/(n-1)$
- $\mathbf{X}\,\mathbf{V} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{U}\,\mathbf{\Sigma}$
- The columns of $\mathbf{U}\,\mathbf{\Sigma}$ are the principal components.

## SVD for dimensionality reduction

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

- Input Data: $\mathbf{X}_{n \times d}$
- Goal: Reduce the dimensionality to $k$ where $k < d$
- Select $k$ first columns of $\mathbf{U}$, and $k \times k$ upper-left part of $\mathbf{\Sigma}$
- Construct $\mathbf{B} = \mathbf{U}_k\,\mathbf{\Sigma}_{k \times k}$
- $\mathbf{B}$ is the required $n \times k$ matrix containing first k PCs.

# Rank-k approximation in the spectral norm

$$\mathbf{X} = \sum_{i=1}^{d} u_i \sigma_i v_i^T$$

The best approximation to $\mathbf{X}$ by a rank deficient matrix is obtained by the top singular values and vectors of $\mathbf{X}$.

$$\mathbf{X}_k = \sum_{i=1}^{k} u_i \sigma_i v_i^T$$

Then,

$$\min_{\mathbf{B} \in \mathbb{R}^{n \times d} \; \mathrm{rank}(\mathbf{B}) \le k} \left\| \mathbf{X} - \mathbf{B} \right\|_2 = \left\| \mathbf{X} - \mathbf{X}_k \right\|_2 = \sigma_{k+1}$$

$\sigma_{k+1}$ is the largest singular value of $\mathbf{X} - \mathbf{X}_k$.
$\mathbf{X}_k$ is the best rank k 2-norm approximation of $\mathbf{X}$.

# Applications of SVD

- Determining range, null space and rank (also numerical rank).
- Matrix approximation.
- Inverse and Pseudo-inverse:
  - If $\mathbf{X} = \mathbf{U} \, \mathbf{\Sigma} \, \mathbf{V}^T$ and $\mathbf{\Sigma}$ is full rank, then $\mathbf{X}^{-1} = \mathbf{V} \, \mathbf{\Sigma}^{-1} \mathbf{U}^T$.
  - If $\mathbf{\Sigma}$ is singular, then its pseudo-inverse is given by $\mathbf{X}^{\dagger} = \mathbf{V} \, \mathbf{\Sigma}^{\dagger} \mathbf{U}^T$ , where $\mathbf{\Sigma}^{\dagger}$ is formed by replacing every nonzero entry by its reciprocal.
- Least squares:
  - If we need to solve $\mathbf{A}\mathbf{x} = \mathrm{b}$ in the least-squares sense, then $\mathbf{x}_{LS} = \mathbf{V} \, \mathbf{\Sigma}^{\dagger} \mathbf{U}^T \, \mathrm{b}$
- Denoising – Small singular values typically correspond to noise.

# Latent Semantic Analysis using SVD

- Input matrix: term-document matrices
  - Rows: represents words.
  - Columns: represents documents.
  - Value: the count of the words in the document.
  - Example:

| |
|---|
| D1:How to bake bread without recipes |
| D2:The classic art of Viennese pastry |
| D3:Numerical recipes: The art of scientific computing |
| D4:Breads, pastries, pies and cakes: quantity baking recipes |
| D5:Pastry: A book of best french recipes |

| T1:bak(e,ing) | T2:recipes | T3:bread | T4:cake | T5:pastr(y,ies) | T6:pie |
|---|---|---|---|---|---|

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

# Latent Semantic Indexing (LSI)

$$\mathbf{X}_{n \times d} = \mathbf{U}_{n \times k} \mathbf{\Sigma}_{k \times k} \mathbf{V}^{T}_{k \times d}$$

- Consider $\mathbf{X}$, the term-document matrix.
- Then,
  - $\mathbf{U}$ is the SVD term matrix
  - $\mathbf{V}$ is the SVD document matrix
- SVD provides a low rank approximation for $\mathbf{X}$.
- Constrained optimization problem
  - Goal: Represent $\mathbf{X}$ as $\mathbf{X}_k$ with low Frobenius norm for the error $\mathbf{X}$ - $\mathbf{X}_k$

# Latent Semantic Indexing (LSI)

$$\mathbf{X}_{n \times d} = \mathbf{U}_{n \times k} \boldsymbol{\Sigma}_{k \times k} \mathbf{V}^{T}_{k \times d}$$

$\mathbf{X} =$

| | d1 | d2 | d3 | d4 | d5 | d6 |
|---|---|---|---|---|---|---|
| cosmonaut | 1 | 0 | 1 | 0 | 0 | 0 |
| astronaut | 0 | 1 | 0 | 0 | 0 | 0 |
| moon | 1 | 1 | 0 | 0 | 0 | 0 |
| car | 1 | 0 | 0 | 1 | 1 | 0 |
| truck | 0 | 0 | 0 | 1 | 0 | 1 |

$\mathbf{U} =$

| | dim 1 | dim 2 | dim 3 | dim 4 | dim 5 |
|---|---|---|---|---|---|
| cosmonaut | −0.44 | −0.30 | 0.57 | 0.58 | 0.25 |
| astronaut | −0.13 | −0.33 | −0.59 | 0.00 | 0.73 |
| moon | −0.48 | −0.51 | −0.37 | 0.00 | −0.61 |
| car | −0.70 | 0.35 | 0.15 | −0.58 | 0.16 |
| truck | −0.26 | 0.65 | −0.41 | 0.58 | −0.09 |

$\boldsymbol{\Sigma} =$

| | | | | |
|---|---|---|---|---|
| 2.16 | 0 | 0 | 0 | 0 |
| 0 | 1.59 | 0 | 0 | 0 |
| 0 | 0 | 1.28 | 0 | 0 |
| 0 | 0 | 0 | 1.00 | 0 |
| 0 | 0 | 0 | 0 | 0.39 |

$\mathbf{V}^{T} =$

| | d1 | d2 | d3 | d4 | d5 | d6 |
|---|---|---|---|---|---|---|
| dim1 | −0.75 | −0.28 | −0.20 | −0.45 | −0.33 | −0.12 |
| dim2 | −0.29 | −0.53 | −0.19 | 0.63 | 0.22 | 0.41 |
| dim3 | 0.28 | −0.75 | 0.45 | −0.20 | 0.12 | −0.33 |
| dim4 | 0 | 0 | 0.58 | 0 | −0.58 | 0.58 |
| dim5 | −0.53 | 0.29 | −0.63 | 0.19 | 0.41 | −0.22 |

# Latent Semantic Indexing (LSI)

$$\mathbf{X}_{n \times d} = \mathbf{U}_{n \times k} \boldsymbol{\Sigma}_{k \times k} \mathbf{V}^{T}_{k \times d}$$

$k = 2$

$\mathbf{X} =$

| | d1 | d2 | d3 | d4 | d5 | d6 |
|---|---|---|---|---|---|---|
| cosmonaut | 1 | 0 | 1 | 0 | 0 | 0 |
| astronaut | 0 | 1 | 0 | 0 | 0 | 0 |
| moon | 1 | 1 | 0 | 0 | 0 | 0 |
| car | 1 | 0 | 0 | 1 | 1 | 0 |
| truck | 0 | 0 | 0 | 1 | 0 | 1 |

$\mathbf{U} =$

| | dim 1 | dim 2 | dim 3 | dim 4 | dim 5 |
|---|---|---|---|---|---|
| cosmonaut | −0.44 | −0.30 | 0 | 0 | 0 |
| astronaut | −0.13 | −0.33 | 0 | 0 | 0 |
| moon | −0.48 | −0.51 | 0 | 0 | 0 |
| car | −0.70 | 0.35 | 0 | 0 | 0 |
| truck | −0.26 | 0.65 | 0 | 0 | 0 |

We can get rid of zero valued columns
And have a 5 x 2 **term-to-concept similarity matrix**

$\boldsymbol{\Sigma} =$

| | | | | |
|---|---|---|---|---|
| 2.16 | 0 | 0 | 0 | 0 |
| 0 | 1.59 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

We can get rid of zero valued columns and rows
And have a 2 x 2 **concept strength matrix**

$\mathbf{V}^{T} =$

| | d1 | d2 | d3 | d4 | d5 | d6 |
|---|---|---|---|---|---|---|
| dim 1 | −0.75 | −0.28 | −0.20 | −0.44 | −0.33 | −0.12 |
| dim 2 | −0.29 | −0.53 | −0.19 | 0.65 | 0.22 | 0.41 |
| dim 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| dim 4 | 0 | 0 | 0 | 0 | 0 | 0 |
| dim 5 | 0 | 0 | 0 | 0 | 0 | 0 |

We can get rid of zero valued columns
And have a 2 x 6 **concept-to-doc similarity matrix**

# Latent Semantic Indexing (LSI)

Original space

$$Q = \begin{pmatrix} cosmonaut & 1 \\ astronaut & 0 \\ moon & 0 \\ car & 0 \\ truck & 0 \end{pmatrix}$$

$$\mathbf{X} = \begin{pmatrix} & d1 & d2 & d3 & d4 & d5 & d6 \\ cosmonaut & 1 & 0 & 1 & 0 & 0 & 0 \\ astronaut & 0 & 1 & 0 & 0 & 0 & 0 \\ moon & 1 & 1 & 0 & 0 & 0 & 0 \\ car & 1 & 0 & 0 & 1 & 1 & 0 \\ truck & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

$$\cos(Q, d2) = 0$$

Reduced latent semantic space

$$Q^r = \begin{pmatrix} \text{dim 1} & -0.44 \\ \text{dim 2} & -0.30 \end{pmatrix}$$

$$\mathbf{V}^T = \begin{pmatrix} & d1 & d2 & d3 & d4 & d5 & d6 \\ \text{dim 1} & -0.75 & -0.28 & -0.20 & -0.44 & -0.33 & -0.12 \\ \text{dim 2} & -0.29 & -0.53 & -0.19 & 0.65 & 0.22 & 0.41 \\ \text{dim 3} & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{dim 4} & 0 & 0 & 0 & 0 & 0 & 0 \\ \text{dim 5} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\cos(Q^r, d2) = 0.88$$

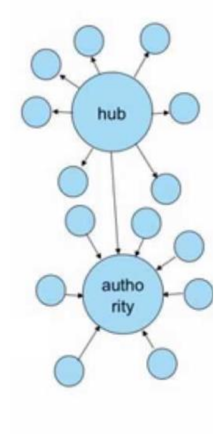
We see that query is not related to document 2 in the original space but in the latent semantic space they become highly related.

# Latent Semantic Indexing (LSI)

- SVD allow words and documents to be mapped into the same "latent semantic space“.
- LSI projects queries and documents into a space with latent semantic dimensions.
  - Co-occurring words are projected on the same dimensions
  - Non-co-occurring words are projected onto different dimensions
- LSI captures similarities between words
  - For example, we want to project “car” and “automobile” onto the same dimension.
- Dimensions of the reduced semantic space correspond to the axes of greatest variation in the original space.

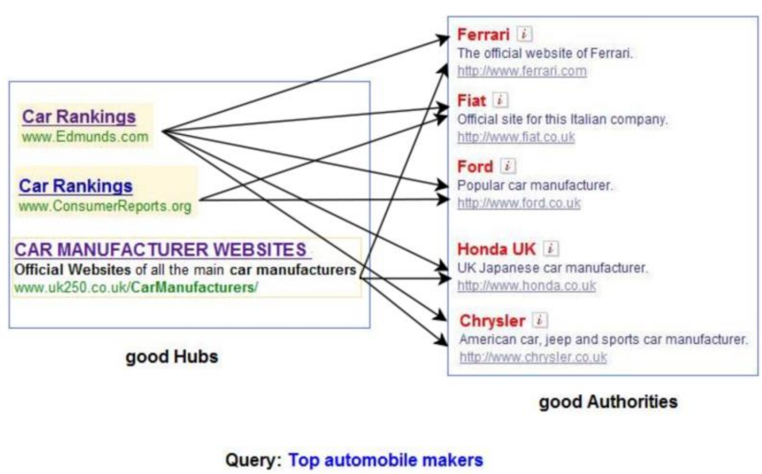
# Kleinberg's Algorithm
# Hyperlink-Induced Topic Search (HITS)
# aka 'hubs and authorities'

- Extracting information from link structures of a hyperlinked environment, rank pages relevant to a topic
- Essentials:
  - Authorities
  - Hubs
- Goal: Identify good authorities and hubs for a topic.
- Each page receive two scores,
  - Authority score $A(p)$: It estimates value of content on page
  - Hub score $H(p)$: It estimates value of links on page

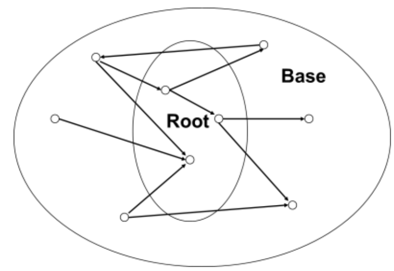



# Authorities and Hubs

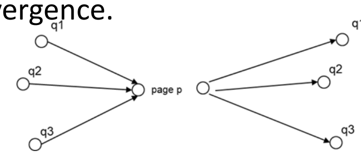
- For a topic, authorities are relevant nodes which are referred by many hubs. (high in degree)
- For a topic, hubs are nodes which connect many related authorities for that topic. (high out degree)



# HITS (cont.)

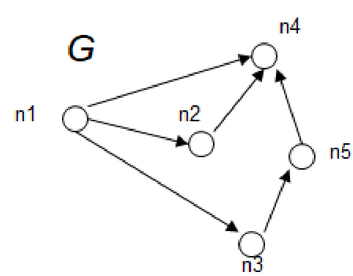



- Three Steps
  1. Create a focused base-set of the Web.
     - Start with a root set.
     - Add any page pointed by a page in the root set to it.
     - Add any page that points to a page in the root set to it (at most d).
     - The extended root set becomes our base set.
  2. Iteratively compute hub and authority scores.
     - $\mathrm{A(p)}$: sum of $\mathrm{H(q)}$ for all $\mathrm{q}$ pointing to $\mathrm{p}$.
     - $\mathrm{H(q)}$: sum of $\mathrm{A(p)}$ for all $\mathrm{p}$ pointing to $\mathrm{q}$.
     - Starts with all scores as 1, and Iteratively repeat till convergence.
  3. Filter out the top hubs and authorities




# Matrix Notation

- G (root set) is a directed graph with web pages as nodes and their links.
- G can be presented as a connectivity matrix $\mathrm{A}$
  - A(i,j)=1 only if i-th page points to j-th page.
- Authority weights can be represented as a unit vector a
  - $a_i$ The authority weight of the i-th page
- Hub weights can be represented as a unit vector $\mathrm{h}$
  - $h_i$ : The hub weight of the i-th page




$$A = \begin{pmatrix} & n1 & n2 & n3 & n4 & n5 \\ n1 & 0 & 1 & 1 & 1 & 0 \\ n2 & 0 & 0 & 0 & 1 & 0 \\ n3 & 0 & 0 & 0 & 0 & 1 \\ n4 & 0 & 0 & 0 & 0 & 0 \\ n5 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

# Algorithm

- Updating authority weights:

$$a = A^T h$$

- Updating hub weights:

$$h = Aa$$

- After k iterations:

$$a_1 = A^T h_0$$
$$h_1 = Aa_1$$
$$\rightarrow h_1 = AA^T h_0$$
$$\rightarrow h_k = (AA^T)_k h_0$$

- Convergence
  - $a_k$: Converges to principal eigen vector of $A^T A$
  - $h_k$: Converges to principal eigen vector of $AA^T$

# Nonnegative Matrix Factorization (NMF)

Given $A \in \mathbb{R}_+^{n \times d}$ and a desired rank $k \ll \min(n, d)$,

Find $W \in \mathbb{R}_+^{n \times k}$ and $H \in \mathbb{R}_+^{k \times n}$ s.t. $A \approx WH$.

- $\min_{W \geq 0, H \geq 0} ||A - WH||_F$
- Nonconvex.
- W and H not unique ( e.g. $\widehat{W} = WD \geq 0, \widehat{H} = D^{-1}H \geq 0$)

Notation: $\mathbb{R}_+$ nonnegative real numbers

## Nonnegative Matrix Factorization (NMF)

- SVD gives: $A = U\Sigma V^T$
  - Then, $\left|\left|A - U\Sigma V^T\right|\right|_F \leq \min||A - WH||_F$
  - Then WHY NMF???

- NMF works better in terms of its non-negativity constraints. Example in
  - Text mining. (A is represented as counts, so is strictly positive.)

## References


- https://ocw.mit.edu/courses/mathematics/18-06sc-linear-algebra-fall-2011/positive-definite-matrices-and-applications/singular-value-decomposition/MIT18_06SCF11_Ses3.5sum.pdf
- https://archive.siam.org/meetings/sdm11/park.pdf