COMPSCI 514: Algorithms for Data Science

Cameron Musco

University of Massachusetts Amherst. Fall 2023.
Lecture 18

Logistics

- Problem Set 3 is due next **Friday at 11:59pm**.
- I made a small change to Problem 1.4: replacing $\sum_{i=1}^{n} \sigma_i(A)^2$ with $\sum_{i=1}^{\text{rank}(A)} \sigma_i(A)^2$. This don't change the solution to the problem, but as we will see will better match the conventions for SVD that I introduce today.

LGRC A311

- Linear algebra review session **Monday 2-3pm**. Location ~~TBD~~.

Summary

Last Class

- Finish up optimal low-rank approximation via eigendecomposition.

$X^T X$

- Eigenvalue spectrum as a way of measuring low-rank approximation error.

This Class: The SVD and Application of Low-Rank Approximation Beyond Compression

- The Singular Value Decomposition (SVD) and its connection to eigendecomposition and low-rank approximation.
- Low-rank matrix completion (predicting missing measurements using low-rank structure).
- Entity embeddings (e.g., word embeddings, node embeddings).
- Low-rank approximation for non-linear dimensionality reduction.

Singular Value Decomposition

The Singular Value Decomposition (SVD) generalizes the eigendecomposition to asymmetric (even rectangular) matrices.

Singular Value Decomposition

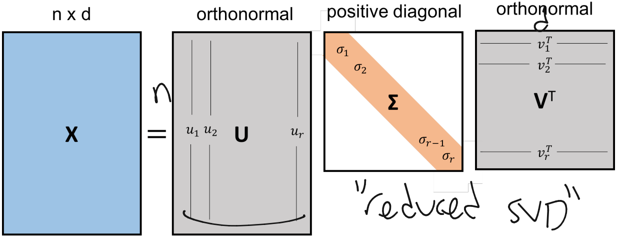
The Singular Value Decomposition (SVD) generalizes the eigendecomposition to asymmetric (even rectangular) matrices. Any matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ with $\text{rank}(\mathbf{X}) = r$ can be written as $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$.

- $\mathbf{U}$ has orthonormal columns $\vec{u}_1, \ldots, \vec{u}_r \in \mathbb{R}^n$ (left singular vectors).
- $\mathbf{V}$ has orthonormal columns $\vec{v}_1, \ldots, \vec{v}_r \in \mathbb{R}^d$ (right singular vectors).
- $\boldsymbol{\Sigma}$ is diagonal with elements $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$ (singular values).

~~$U \in \mathbb{R}^{n\times n}$ $\Sigma^{n\times d}$ $V \in \mathbb{R}^{d\times d}$~~

r

$X = V\Lambda V^T$



"reduced SVD"

Connection of the SVD to Eigendecomposition

Writing $\mathbf{X} \in \mathbb{R}^{n \times d}$ in its singular value decomposition $\underline{\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T}$:

$$\underline{\mathbf{X}^T\mathbf{X} =} \ (U\Sigma V^T)^T (U\Sigma V^T)$$

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

$$V\Sigma\Sigma V^T = V\Sigma^2 V^T$$

$\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\mathbf{U} \in \mathbb{R}^{n \times \text{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{u}_1, \vec{u}_2, \ldots$ (left singular vectors), $\mathbf{V} \in \mathbb{R}^{d \times \text{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{v}_1, \vec{v}_2, \ldots$ (right singular vectors), $\mathbf{\Sigma} \in \mathbb{R}^{\text{rank}(\mathbf{X}) \times \text{rank}(\mathbf{X})}$: positive diagonal matrix containing singular values of $\mathbf{X}$.

Connection of the SVD to Eigendecomposition

Writing $\mathbf{X} \in \mathbb{R}^{n \times d}$ in its singular value decomposition $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$:

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

$\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\mathbf{U} \in \mathbb{R}^{n \times \text{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{u}_1, \vec{u}_2, \ldots$ (left singular vectors), $\mathbf{V} \in \mathbb{R}^{d \times \text{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{v}_1, \vec{v}_2, \ldots$ (right singular vectors), $\boldsymbol{\Sigma} \in \mathbb{R}^{\text{rank}(\mathbf{X}) \times \text{rank}(\mathbf{X})}$: positive diagonal matrix containing singular values of $\mathbf{X}$.

Connection of the SVD to Eigendecomposition

Writing $\mathbf{X} \in \mathbb{R}^{n \times d}$ in its singular value decomposition $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$:

$$\mathbf{X}^T\mathbf{X} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \mathbf{V}\underline{\boldsymbol{\Sigma}^2}\mathbf{V}^T$$

$\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\mathbf{U} \in \mathbb{R}^{n \times \text{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{u}_1, \vec{u}_2, \ldots$ (left singular vectors), $\mathbf{V} \in \mathbb{R}^{d \times \text{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{v}_1, \vec{v}_2, \ldots$ (right singular vectors), $\boldsymbol{\Sigma} \in \mathbb{R}^{\text{rank}(\mathbf{X}) \times \text{rank}(\mathbf{X})}$: positive diagonal matrix containing singular values of $\mathbf{X}$.

Connection of the SVD to Eigendecomposition

Writing $\mathbf{X} \in \mathbb{R}^{n \times d}$ in its singular value decomposition $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$:

$$\mathbf{X}^T\mathbf{X} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \mathbf{V}\boldsymbol{\Sigma}^2\mathbf{V}^T \text{ (the eigendecomposition)}$$

$\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\mathbf{U} \in \mathbb{R}^{n \times \text{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{u}_1, \vec{u}_2, \ldots$ (left singular vectors), $\mathbf{V} \in \mathbb{R}^{d \times \text{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{v}_1, \vec{v}_2, \ldots$ (right singular vectors), $\boldsymbol{\Sigma} \in \mathbb{R}^{\text{rank}(\mathbf{X}) \times \text{rank}(\mathbf{X})}$: positive diagonal matrix containing singular values of $\mathbf{X}$.

Connection of the SVD to Eigendecomposition

Writing $\mathbf{X} \in \mathbb{R}^{n \times d}$ in its singular value decomposition $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$:

$$\mathbf{X}^T\mathbf{X} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{U}\boldsymbol{\Sigma}\underset{I}{\mathbf{V}}^T = \mathbf{V}\boldsymbol{\Sigma}^2\mathbf{V}^T \text{ (the eigendecomposition)}$$

Similarly: $\mathbf{X}\mathbf{X}^T = \underbrace{\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T}\mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T = \mathbf{U}\boldsymbol{\Sigma}^2\mathbf{U}^T$.

$\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\mathbf{U} \in \mathbb{R}^{n \times \text{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{u}_1, \vec{u}_2, \ldots$ (left singular vectors), $\mathbf{V} \in \mathbb{R}^{d \times \text{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{v}_1, \vec{v}_2, \ldots$ (right singular vectors), $\boldsymbol{\Sigma} \in \mathbb{R}^{\text{rank}(\mathbf{X}) \times \text{rank}(\mathbf{X})}$: positive diagonal matrix containing singular values of $\mathbf{X}$.

Connection of the SVD to Eigendecomposition

Writing $\mathbf{X} \in \mathbb{R}^{n \times d}$ in its singular value decomposition $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$:

$$\mathbf{X}^T\mathbf{X} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \mathbf{V}\boldsymbol{\Sigma}^2\mathbf{V}^T \text{ (the eigendecomposition)}$$

Similarly: $\mathbf{X}\mathbf{X}^T = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T = \mathbf{U}\boldsymbol{\Sigma}^2\mathbf{U}^T$.

The left and right singular vectors are the eigenvectors of the covariance matrix $\mathbf{X}^T\mathbf{X}$ and the gram matrix $\mathbf{X}\mathbf{X}^T$ respectively.

$$\sigma_i(X)^2 = \lambda_i(X^TX) = \lambda_i(XX^T)$$

$$n\left[\widehat{XX^T}\right]_{ij} = \langle x_i, x_j \rangle$$

*

$$\lambda_i(AB) = \lambda_i(BA)$$

$$\det(A - \lambda I) = 0$$

$$(A - \lambda I)v$$

$$Av - \lambda v$$

$$\lambda v - \lambda v = 0$$

$\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\mathbf{U} \in \mathbb{R}^{n \times \text{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{u}_1, \vec{u}_2, \ldots$ (left singular vectors), $\mathbf{V} \in \mathbb{R}^{d \times \text{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{v}_1, \vec{v}_2, \ldots$ (right singular vectors), $\boldsymbol{\Sigma} \in \mathbb{R}^{\text{rank}(\mathbf{X}) \times \text{rank}(\mathbf{X})}$: positive diagonal matrix containing singular values of $\mathbf{X}$.

Connection of the SVD to Eigendecomposition

Writing $\mathbf{X} \in \mathbb{R}^{n \times d}$ in its singular value decomposition $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$:

$$\mathbf{X}^T\mathbf{X} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \mathbf{V}\boldsymbol{\Sigma}^2\mathbf{V}^T \text{ (the eigendecomposition)}$$

Similarly: $\mathbf{X}\mathbf{X}^T = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T = \mathbf{U}\boldsymbol{\Sigma}^2\mathbf{U}^T$.

The left and right singular vectors are the eigenvectors of the covariance matrix $\mathbf{X}^T\mathbf{X}$ and the gram matrix $\mathbf{X}\mathbf{X}^T$ respectively. X^TX

So, letting $\mathbf{V}_k \in \mathbb{R}^{d \times k}$ have columns equal to $\vec{v}_1, \ldots, \vec{v}_k$, we know that $\mathbf{X}\mathbf{V}_k\mathbf{V}_k^T$ is the best rank-k approximation to $\mathbf{X}$ (given by PCA).

eigenvecs of X^TX

$\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\mathbf{U} \in \mathbb{R}^{n \times \mathrm{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{u}_1, \vec{u}_2, \ldots$ (left singular vectors), $\mathbf{V} \in \mathbb{R}^{d \times \mathrm{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{v}_1, \vec{v}_2, \ldots$ (right singular vectors), $\boldsymbol{\Sigma} \in \mathbb{R}^{\mathrm{rank}(\mathbf{X}) \times \mathrm{rank}(\mathbf{X})}$: positive diagonal matrix containing singular values of $\mathbf{X}$.

Connection of the SVD to Eigendecomposition

Writing $\mathbf{X} \in \mathbb{R}^{n \times d}$ in its singular value decomposition $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$:

$$\mathbf{X}^T\mathbf{X} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \mathbf{V}\boldsymbol{\Sigma}^2\mathbf{V}^T \text{ (the eigendecomposition)}$$

Similarly: $\mathbf{X}\mathbf{X}^T = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T = \mathbf{U}\boldsymbol{\Sigma}^2\mathbf{U}^T$.

The left and right singular vectors are the eigenvectors of the covariance matrix $\mathbf{X}^T\mathbf{X}$ and the gram matrix $\mathbf{X}\mathbf{X}^T$ respectively.

So, letting $\mathbf{V}_k \in \mathbb{R}^{d \times k}$ have columns equal to $\vec{v}_1, \ldots, \vec{v}_k$, we know that $\underline{\mathbf{X}\mathbf{V}_k\mathbf{V}_k^T}$ is the best rank-k approximation to $\mathbf{X}$ (given by PCA).

What about $\mathbf{U}_k\mathbf{U}_k^T\mathbf{X}$ where $\mathbf{U}_k \in \mathbb{R}^{n \times k}$ has columns equal to $\vec{u}_1, \ldots, \vec{u}_k$?

$\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\mathbf{U} \in \mathbb{R}^{n \times \text{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{u}_1, \vec{u}_2, \ldots$ (left singular vectors), $\mathbf{V} \in \mathbb{R}^{d \times \text{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{v}_1, \vec{v}_2, \ldots$ (right singular vectors), $\boldsymbol{\Sigma} \in \mathbb{R}^{\text{rank}(\mathbf{X}) \times \text{rank}(\mathbf{X})}$: positive diagonal matrix containing singular values of $\mathbf{X}$.

Connection of the SVD to Eigendecomposition

Writing $\mathbf{X} \in \mathbb{R}^{n \times d}$ in its singular value decomposition $\mathbf{X} = \underline{\mathbf{U}}\mathbf{\Sigma}\underline{\mathbf{V}^T}$:

$$\underline{\mathbf{X}^T\mathbf{X}} = \mathbf{V}\mathbf{\Sigma}\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \underline{\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T} \text{ (the eigendecomposition)}$$

Similarly: $\mathbf{X}\mathbf{X}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{\Sigma}\mathbf{U}^T = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T$.

The left and right singular vectors are the eigenvectors of the covariance matrix $\mathbf{X}^T\mathbf{X}$ and the gram matrix $\mathbf{X}\mathbf{X}^T$ respectively.

V_k top k eigvecs of X^TX

So, letting $\mathbf{V}_k \in \mathbb{R}^{d \times k}$ have columns equal to $\vec{v}_1, \ldots, \vec{v}_k$, we know that $\mathbf{X}\mathbf{V}_k\mathbf{V}_k^T$ is the best rank-k approximation to $\mathbf{X}$ (given by PCA).

$XV_kV_k^T$ best LRA

What about $\mathbf{U}_k\mathbf{U}_k^T\mathbf{X}$ where $\mathbf{U}_k \in \mathbb{R}^{n \times k}$ has columns equal to $\vec{u}_1, \ldots, \vec{u}_k$?

Gives exactly the same approximation!

$$XV_kV_k^T = U_kU_k^TX$$

X

$\mathbf{X} \in \mathbb{R}^{n \times d}$: data matrix, $\mathbf{U} \in \mathbb{R}^{n \times \text{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{u}_1, \vec{u}_2, \ldots$ (left singular vectors), $\mathbf{V} \in \mathbb{R}^{d \times \text{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{v}_1, \vec{v}_2, \ldots$ (right singular vectors), $\mathbf{\Sigma} \in \mathbb{R}^{\text{rank}(\mathbf{X}) \times \text{rank}(\mathbf{X})}$: positive diagonal matrix containing singular values of $\mathbf{X}$.

The SVD and Optimal Low-Rank Approximation

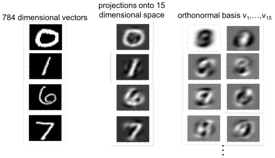
The best low-rank approximation to $\mathbf{X}$:

$\mathbf{X}_k = \arg\min_{\text{rank}-k\ \mathbf{B}\in\mathbb{R}^{n\times d}} \|\underline{\mathbf{X}-\mathbf{B}}\|_F$ is given by:

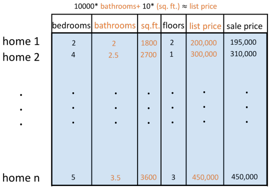
$$\underline{\mathbf{X}_k = \underline{\mathbf{X}\mathbf{V}_k\mathbf{V}_k^T} = \underline{\mathbf{U}_k\mathbf{U}_k^T\mathbf{X}}}$$

Correspond to projecting the rows (data points) onto the span of $\mathbf{V}_k$ or the columns (features) onto the span of $\mathbf{U}_k$

Row (data point) compression


784 dimensional vectors
projections onto 15 dimensional space
orthonormal basis $v_1,\ldots,v_{15}$

Column (feature) compression



10000* bathrooms+ 10* (sq. ft.) ≈ list price

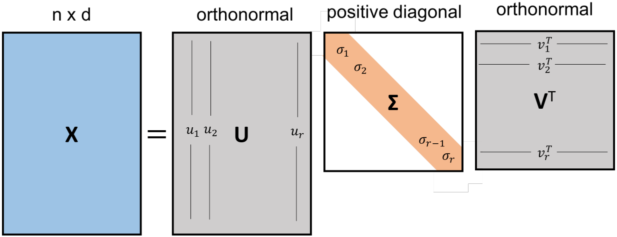
	bedrooms	bathrooms	sq.ft.	floors	list price	sale price
home 1	2	2	1800	2	200,000	195,000
home 2	4	2.5	2700	1	300,000	310,000
⋮	⋮	⋮	⋮	⋮	⋮	⋮
home n	5	3.5	3600	3	450,000	450,000

The SVD and Optimal Low-Rank Approximation

The best low-rank approximation to $\mathbf{X}$:
$\mathbf{X}_k = \arg\min_{\text{rank}-k\ \mathbf{B}\in\mathbb{R}^{n\times d}} \|\mathbf{X}-\mathbf{B}\|_F$ is given by:

$$\mathbf{X}_k = \mathbf{X}\mathbf{V}_k\mathbf{V}_k^T = \mathbf{U}_k\mathbf{U}_k^T\mathbf{X}$$

Correspond to projecting the rows (data points) onto the span of $\mathbf{V}_k$ or the columns (features) onto the span of $\mathbf{U}_k$



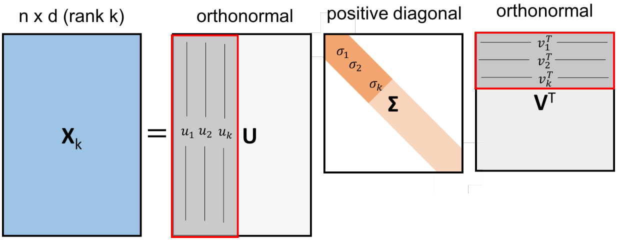
The SVD and Optimal Low-Rank Approximation

The best low-rank approximation to $\mathbf{X}$:

$\mathbf{X}_k = \arg\min_{\text{rank }-k\ \mathbf{B}\in\mathbb{R}^{n\times d}} \|\mathbf{X}-\mathbf{B}\|_F$ is given by:

$$\mathbf{X}_k = \mathbf{X}\mathbf{V}_k\mathbf{V}_k^T = \mathbf{U}_k\mathbf{U}_k^T\mathbf{X}$$

Correspond to projecting the rows (data points) onto the span of $\mathbf{V}_k$ or the columns (features) onto the span of $\mathbf{U}_k$



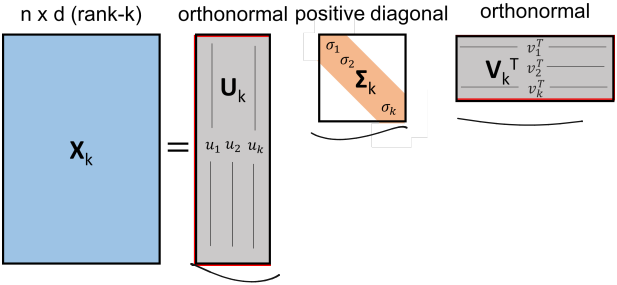
The SVD and Optimal Low-Rank Approximation

The best low-rank approximation to $\mathbf{X}$:

$\mathbf{X}_k = \arg\min_{\text{rank}-k\ \mathbf{B}\in\mathbb{R}^{n\times d}} \|\mathbf{X}-\mathbf{B}\|_F$ is given by:

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

Correspond to projecting the rows (data points) onto the span of $\mathbf{V}_k$ or the columns (features) onto the span of $\mathbf{U}_k$



The SVD and Optimal Low-Rank Approximation

The best low-rank approximation to X:

$X_k = \arg\min_{\text{rank}-k\ \mathbf{B}\in\mathbb{R}^{n\times d}} \|\mathbf{X}-\mathbf{B}\|_F$ is given by:

$$r\begin{bmatrix} \overset{r}{\Sigma} \end{bmatrix} \rightarrow k\begin{bmatrix} \underset{k}{\Sigma_k} \end{bmatrix}$$

first k colums of Σ

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

$$XV_kV_k^T = U\Sigma \underbrace{V^TV_k}V_k^T \longrightarrow U\Sigma I_kV_k^T$$

$$r\begin{bmatrix} \overset{d}{v_1^T} \\ v_2^T \\ \vdots \\ v_r^T \end{bmatrix}\begin{bmatrix} | & & | \\ v_1 & \dots & v_k \\ | & & | \end{bmatrix} = r\begin{bmatrix} \overset{k}{I_{k\times k}} \\ \text{- - - -} \\ 0 \end{bmatrix}$$

$\boxed{I_k}$

$$U\ r\begin{bmatrix} \overset{k}{\Sigma_k} \\ 0 \end{bmatrix} V_k^T$$

$$\begin{bmatrix} u_1 \dots u_r \end{bmatrix}\begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \\ & 0 & \end{bmatrix} = U_k\Sigma_k$$

$$U_k\Sigma_{kk}V_k^T$$

$\mathbf{X} \in \mathbb{R}^{n\times d}$: data matrix, $\mathbf{U} \in \mathbb{R}^{n\times \text{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{u}_1, \vec{u}_2, \dots$ (left singular vectors), $\mathbf{V} \in \mathbb{R}^{d\times \text{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{v}_1, \vec{v}_2, \dots$ (right singular vectors), $\mathbf{\Sigma} \in \mathbb{R}^{\text{rank}(\mathbf{X})\times \text{rank}(\mathbf{X})}$: positive diagonal matrix containing singular values of $\mathbf{X}$.

The SVD and Optimal Low-Rank Approximation

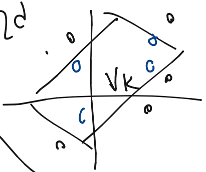
~~$n \geq d$?~~ $\boxed{n \geq k}$

The best low-rank approximation to $\mathbf{X}$:

$\mathbf{X}_k = \arg\min_{\text{rank}-k\ \mathbf{B}\in\mathbb{R}^{n\times d}}^{\text{rank}(B)=k} \|\mathbf{X}-\mathbf{B}\|_F$ is given by:

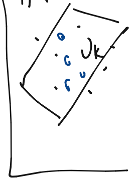
$$\mathbf{X}_k = \boxed{\mathbf{X}\mathbf{V}_k\mathbf{V}_k^T = \mathbf{U}_k\mathbf{U}_k^T\mathbf{X}} = \mathbf{U}_k\boldsymbol{\Sigma}_k\mathbf{V}_k^T$$

rows: $\mathbb{R}^d$



$$\begin{bmatrix} & U & \\ u_1 & \cdots & u_r \\ & & \end{bmatrix}\begin{bmatrix}\sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_k \\ & 0 & & \end{bmatrix} = \begin{bmatrix} | & | & & | \\ u_1\sigma_1 & u_2\sigma_2 & \cdots & u_k\sigma_k \\ | & | & & | \end{bmatrix} = U_k\Sigma_k$$

colums: $\mathbb{R}^n$



$\mathbf{X} \in \mathbb{R}^{n\times d}$: data matrix, $\mathbf{U} \in \mathbb{R}^{n\times \text{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{u}_1, \vec{u}_2, \ldots$ (left singular vectors), $\mathbf{V} \in \mathbb{R}^{d\times \text{rank}(\mathbf{X})}$: matrix with orthonormal columns $\vec{v}_1, \vec{v}_2, \ldots$ (right singular vectors), $\boldsymbol{\Sigma} \in \mathbb{R}^{\text{rank}(\mathbf{X})\times\text{rank}(\mathbf{X})}$: positive diagonal matrix containing singular values of $\mathbf{X}$.

SVD Review

$\mathrm{SVD}(X^TX) = \mathrm{EIG}(X^TX) \qquad X = U\Sigma V^T$

$\downarrow$

$V\Sigma^2V^T$

$\shortparallel$

$V\Lambda V^T$

$(X^TX)(X^TX)^T = (X^TX)^2 = V\Sigma^4V^T$

- Every $\mathbf{X} \in \mathbb{R}^{n\times d}$ can be written in its SVD as $\mathbf{U\Sigma V}^T$.

  $\mathbb{R}^n$

  $n\times n$

- $\mathbf{U} \in \mathbb{R}^{n\times r}$ (orthonormal) contains the eigenvectors of $\mathbf{XX}^T$.
  $\mathbf{V} \in \mathbb{R}^{d\times r}$ (orthonormal) contains the eigenvectors of $\mathbf{X^TX}$.
  $\mathbf{\Sigma}^2 \in \mathbb{R}^{r\times r}$ (diagonal) contains their eigenvalues.

- $\mathbf{U}_k\mathbf{U}_k^T\mathbf{X} = \mathbf{XV}_k\mathbf{V}_k^T = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T = \underset{\mathbf{B} \text{ s.t. } \operatorname{rank}(\mathbf{B})\le k}{\arg\min} \|\mathbf{X}-\mathbf{B}\|_F$.

$$n\overset{n}{\begin{bmatrix} | & & & \\ u_1 & u & \cdot & u_n \\ | & & & \end{bmatrix}}$$

Applications of Low-Rank Approximation Beyond Compression

Matrix Completion

Consider a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ which we cannot fully observe but believe is close to rank-k (i.e., well approximated by a rank k matrix).

Matrix Completion

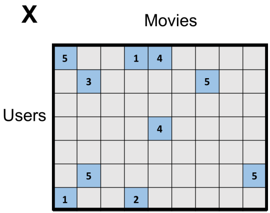
Consider a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ which we cannot fully observe but believe is close to rank-k (i.e., well approximated by a rank k matrix). Classic example: the Netflix prize problem.

$$\mathbf{X} \qquad \begin{array}{c|ccccccccc|}
 & \multicolumn{9}{c}{\text{Movies}} \\
\hline
 & 5 & 3 & 3 & 1 & 4 & 4 & 4 & 3 & 5 \\
 & 4 & 3 & 3 & 1 & 4 & 4 & 5 & 3 & 5 \\
 & 3 & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 3 \\
\text{Users} & 4 & 3 & 3 & 4 & 4 & 4 & 4 & 3 & 3 \\
 & 3 & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 3 \\
 & 2 & \textcircled{5} & 3 & 4 & 4 & 4 & 4 & 4 & 5 \\
 & 1 & 3 & 3 & 2 & 3 & 3 & 3 & 1 & 2 \\
\hline
\end{array}$$

Matrix Completion

Consider a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ which we cannot fully observe but believe is close to rank-k (i.e., well approximated by a rank k matrix). Classic example: the Netflix prize problem.

Why might X be close to low rank?

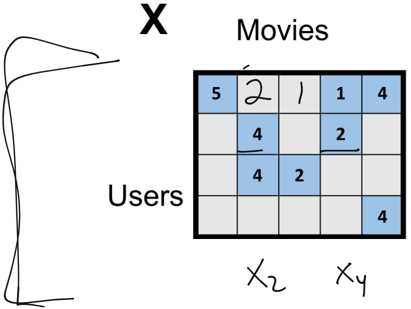


5			1	4				
	3					5		
				4				
	5							5
1			2					

- users similar
- total ratings on movies
- genres

Matrix Completion

Consider a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ which we cannot fully observe but believe is close to rank-k (i.e., well approximated by a rank k matrix). Classic example: the Netflix prize problem.



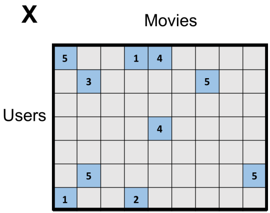
5	2	1	1	4
	4		2	
	4	2		
				4

Assume rank(**X**)=1

$$\begin{bmatrix} 5 & 4 & 3 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Matrix Completion

Consider a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ which we cannot fully observe but believe is close to rank-k (i.e., well approximated by a rank k matrix). Classic example: the Netflix prize problem.

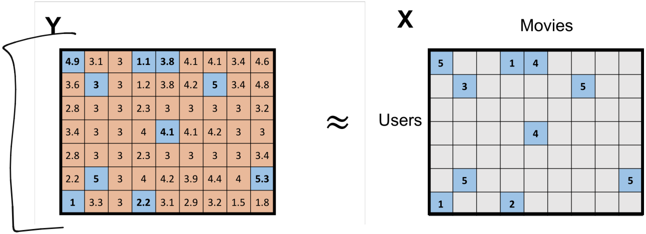


5			1	4				
	3					5		
				4				
	5							5
1			2					

$$\text{Solve: } Y = \underbrace{\mathop{\arg\min}_{\mathbf{B} \text{ s.t. } \mathrm{rank}(\mathbf{B}) \leq k} \sum_{\text{observed } (j,k)} \left[\mathbf{X}_{j,k} - \mathbf{B}_{j,k}\right]^2}$$

Matrix Completion

Consider a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ which we cannot fully observe but believe is close to rank-k (i.e., well approximated by a rank k matrix). Classic example: the Netflix prize problem.



Y

4.9	3.1	3	**1.1**	**3.8**	4.1	4.1	3.4	4.6
3.6	**3**	3	1.2	3.8	4.2	**5**	3.4	4.8
2.8	3	3	2.3	3	3	3	3	3.2
3.4	3	3	4	**4.1**	4.1	4.2	3	3
2.8	3	3	2.3	3	3	3	3	3.4
2.2	**5**	3	4	4.2	3.9	4.4	4	**5.3**
1	3.3	3	**2.2**	3.1	2.9	3.2	1.5	1.8

$\approx$

X

Movies

Users

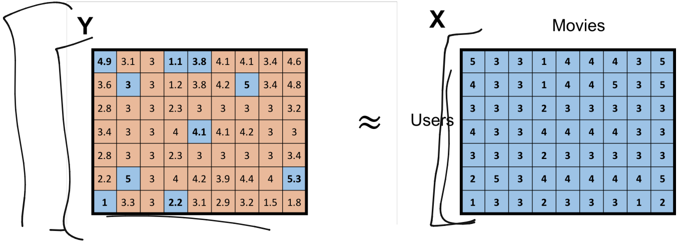
5			**1**	**4**				
	3					**5**		
				4				
	5							**5**
1			**2**					

Solve: $Y = \underset{\mathbf{B} \text{ s.t. } \operatorname{rank}(\mathbf{B}) \leq k}{\arg\min} \sum_{\text{observed } (j,k)} \left[\mathbf{X}_{j,k} - \mathbf{B}_{j,k}\right]^2$

Matrix Completion

Consider a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ which we cannot fully observe but believe is close to rank-k (i.e., well approximated by a rank k matrix). Classic example: the Netflix prize problem.

$\mathrm{rank}(X) \leq \min(n, d)$



Y

4.9	3.1	3	**1.1**	**3.8**	4.1	4.1	3.4	4.6
3.6	**3**	3	1.2	3.8	4.2	**5**	3.4	4.8
2.8	3	3	2.3	3	3	3	3	3.2
3.4	3	3	4	**4.1**	4.1	4.2	3	3
2.8	3	3	2.3	3	3	3	3	3.4
2.2	**5**	3	4	4.2	3.9	4.4	4	**5.3**
1	3.3	3	**2.2**	3.1	2.9	3.2	1.5	1.8

$\approx$

X

Movies

Users

5	3	3	1	4	4	4	3	5
4	3	3	1	4	4	5	3	5
3	3	3	2	3	3	3	3	3
4	3	3	4	4	4	4	3	3
3	3	3	2	3	3	3	3	3
2	5	3	4	4	4	4	4	5
1	3	3	2	3	3	3	1	2

Solve: $$Y = \arg\min_{\mathbf{B} \text{ s.t. } \mathrm{rank}(\mathbf{B}) \leq k} \sum_{\text{observed } (j,k)} \left[\mathbf{X}_{j,k} - \mathbf{B}_{j,k}\right]^2$$

Under certain assumptions, can show that **Y** well approximates **X** on both the observed and (most importantly) unobserved entries.

Entity Embeddings

Dimensionality reduction embeds d-dimensional vectors into k dimensions. But what about when you want to embed objects other than vectors?

- Documents (for topic-based search and classification)
- Words (to identify synonyms, translations, etc.)
- Nodes in a social network

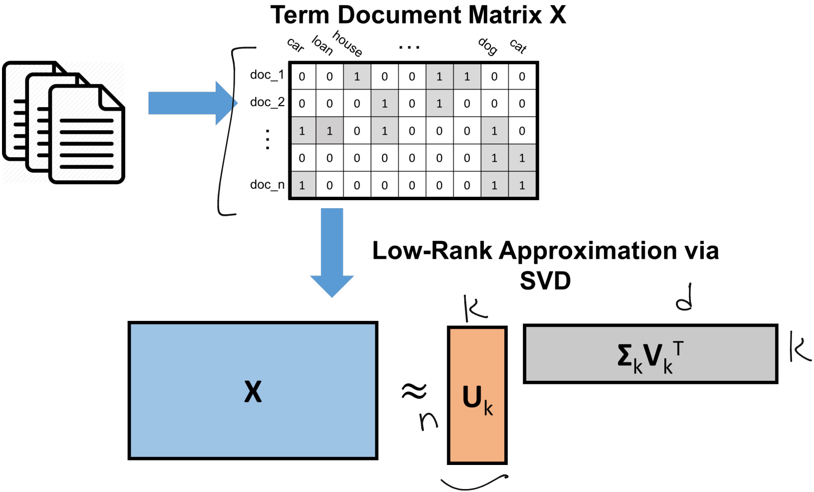
Entity Embeddings

Dimensionality reduction embeds d-dimensional vectors into k dimensions. But what about when you want to embed objects other than vectors?

- Documents (for topic-based search and classification)
- Words (to identify synonyms, translations, etc.)
- Nodes in a social network

Classic Approach: Convert each item into a (very) high-dimensional feature vector and then apply low-rank approximation.

Example: Latent Semantic Analysis

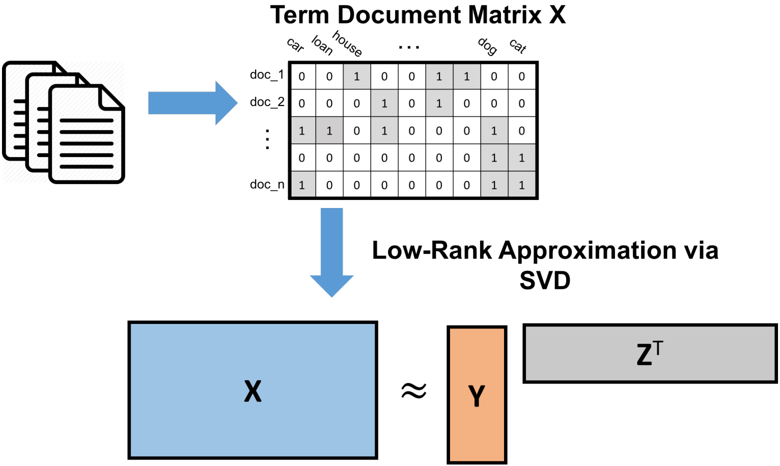


Term Document Matrix X

	car	loan	house		…			dog	cat
doc_1	0	0	1	0	0	1	1	0	0
doc_2	0	0	0	1	0	1	0	0	0
⋮	1	1	0	1	0	0	0	1	0
	0	0	0	0	0	0	0	1	1
doc_n	1	0	0	0	0	0	0	1	1

Low-Rank Approximation via SVD

Example: Latent Semantic Analysis

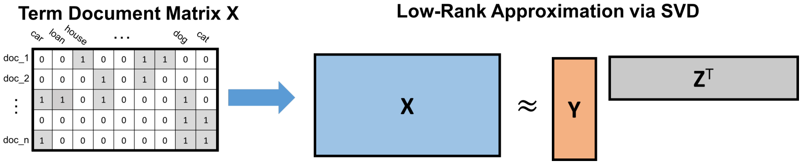


Term Document Matrix X

	car	loan	house		. . .			dog	cat
doc_1	0	0	1	0	0	1	1	0	0
doc_2	0	0	0	1	0	1	0	0	0
⋮	1	1	0	1	0	0	0	1	0
	0	0	0	0	0	0	0	1	1
doc_n	1	0	0	0	0	0	0	1	1

Low-Rank Approximation via SVD

Example: Latent Semantic Analysis

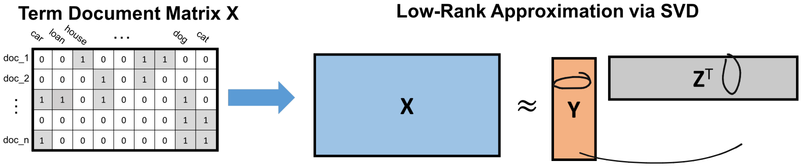


Term Document Matrix X

	car	loan	house		…			dog	cat
doc_1	0	0	1	0	0	1	1	0	0
doc_2	0	0	0	1	0	1	0	0	0
⋮	1	1	0	1	0	0	0	1	0
	0	0	0	0	0	0	0	1	1
doc_n	1	0	0	0	0	0	0	1	1

Low-Rank Approximation via SVD

Example: Latent Semantic Analysis



Term Document Matrix X

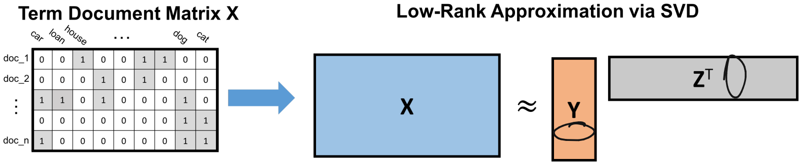
	car	loan	house		…			dog	cat
doc_1	0	0	1	0	0	1	1	0	0
doc_2	0	0	0	1	0	1	0	0	0
⋮	1	1	0	1	0	0	0	1	0
	0	0	0	0	0	0	0	1	1
doc_n	1	0	0	0	0	0	0	1	1

Low-Rank Approximation via SVD

- If the error $\|\underline{X - YZ^T}\|_F$ is small, then on average,

$$\underline{X_{i,a}} \approx \underline{(YZ^T)}_{i,a} = \langle \underline{\vec{y}_i, \vec{z}_a} \rangle.$$

Example: Latent Semantic Analysis



	car	loan	house		…			dog	cat
doc_1	0	0	1	0	0	1	1	0	0
doc_2	0	0	0	1	0	1	0	0	0
⋮	1	1	0	1	0	0	0	1	0
	0	0	0	0	0	0	0	1	1
doc_n	1	0	0	0	0	0	0	1	1

- If the error $\|X - YZ^T\|_F$ is small, then on average,

$$X_{i,a} \approx (YZ^T)_{i,a} = \langle \vec{y}_i, \vec{z}_a \rangle.$$

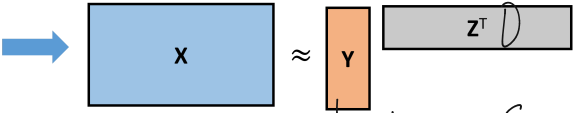
- I.e., $\langle \vec{y}_i, \vec{z}_a \rangle \approx 1$ when doc_i contains $word_a$.

Example: Latent Semantic Analysis

Term Document Matrix X

	car	loan	house		⋯			dog	cat
doc_1	0	0	1	0	0	1	1	0	0
doc_2	0	0	0	1	0	1	0	0	0
⋮	1	1	0	1	0	0	0	1	0
	0	0	0	0	0	0	0	1	1
doc_n	1	0	0	0	0	0	0	1	1

Low-Rank Approximation via SVD



each row of y is k-dim representation of doc

- If the error $\|\mathbf{X} - \mathbf{Y}\mathbf{Z}^T\|_F$ is small, then on average,

$$\mathbf{X}_{i,a} \approx (\mathbf{Y}\mathbf{Z}^T)_{i,a} = \langle \vec{y}_i, \vec{z}_a \rangle.$$

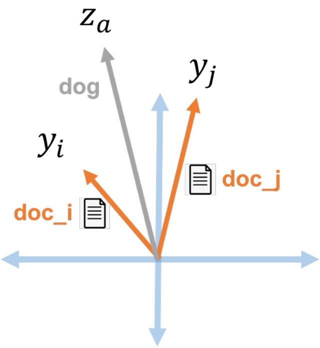
each column Z is
'' of word

- I.e., $\langle \vec{y}_i, \vec{z}_a \rangle \approx 1$ when doc_i contains $word_a$.
- If doc_i and doc_j both contain $word_a$, $\underline{\langle \vec{y}_i, \vec{z}_a \rangle} \approx \underline{\langle \vec{y}_j, \vec{z}_a \rangle} \approx 1$.

$y_i \approx y_j$

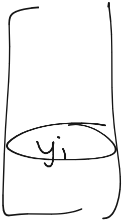
Example: Latent Semantic Analysis

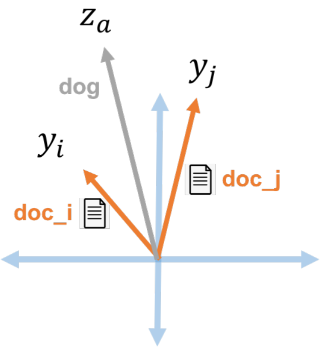
If doc_i and doc_j both contain $word_a$, $\langle \vec{y}_i, \vec{z}_a \rangle \approx \langle \vec{y}_j, \vec{z}_a \rangle \approx 1$



Example: Latent Semantic Analysis

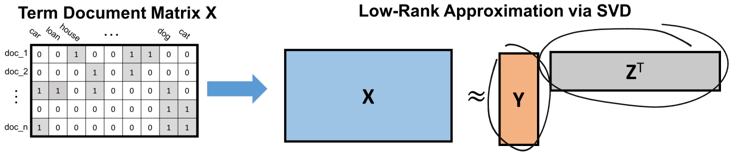
If *doc*$_i$ and *doc*$_j$ both contain *word*$_a$, $\langle \vec{y}_i, \vec{z}_a \rangle \approx \langle \vec{y}_j, \vec{z}_a \rangle \approx 1$





Another View: Each column of **Y** represents a 'topic'. $\vec{y}_i(j)$ indicates how much *doc*$_i$ belongs to topic j. $\vec{z}_a(j)$ indicates how much *word*$_a$ associates with that topic.

Example: Latent Semantic Analysis



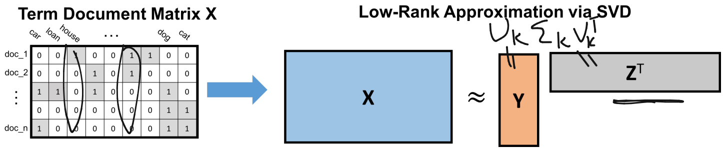
Term Document Matrix X

	car	loan	house		…			dog	cat
doc_1	0	0	1	0	0	1	1	0	0
doc_2	0	0	0	1	0	1	0	0	0
⋮	1	1	0	1	0	0	0	1	0
	0	0	0	0	0	0	0	1	1
doc_n	1	0	0	0	0	0	0	1	1

Low-Rank Approximation via SVD

- Just like with documents, $\vec{z}_a$ and $\vec{z}_b$ will tend to have high dot product if $word_a$ and $word_b$ appear in many of the same documents.

Example: Latent Semantic Analysis



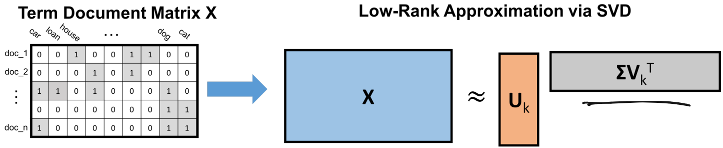
	car	loan	house		…			dog	cat
doc_1	0	0	1	0	0	1	1	0	0
doc_2	0	0	0	1	0	1	0	0	0
⋮	1	1	0	1	0	0	0	1	0
	0	0	0	0	0	0	0	1	1
doc_n	1	0	0	0	0	0	0	1	1

- Just like with documents, $\vec{z}_a$ and $\vec{z}_b$ will tend to have high dot product if $word_a$ and $word_b$ appear in many of the same documents.
- In an SVD decomposition we set $\mathbf{Z}^T = \mathbf{\Sigma}_k \mathbf{V}_K^T$.
- The columns of $\underline{\mathbf{V}_k}$ are equivalently: the top k eigenvectors of $\mathbf{X}^T\mathbf{X}$.

# words appear in both docs

$$\text{docs}\begin{bmatrix} X^T \end{bmatrix} \overset{\text{docs}}{\begin{bmatrix} X \end{bmatrix}} = \begin{bmatrix} (X^TX)_{ij} \end{bmatrix}$$

Example: Latent Semantic Analysis



	car	loan	house		⋯			dog	cat
doc_1	0	0	1	0	0	1	1	0	0
doc_2	0	0	0	1	0	1	0	0	0
⋮	1	1	0	1	0	0	0	1	0
	0	0	0	0	0	0	0	1	1
doc_n	1	0	0	0	0	0	0	1	1

- Just like with documents, $\vec{z}_a$ and $\vec{z}_b$ will tend to have high dot product if $word_a$ and $word_b$ appear in many of the same documents.
- In an SVD decomposition we set $\mathbf{Z}^T = \mathbf{\Sigma}_k \mathbf{V}_K^T$.
- The columns of $\mathbf{V}_k$ are equivalently: the top k eigenvectors of $\mathbf{X}^T\mathbf{X}$.
- **Claim:** $\underline{\mathbf{Z}\mathbf{Z}^T}$ is the best rank-k approximation of $\mathbf{X}^T\mathbf{X}$. I.e., $\arg\min_{\text{rank }-k\ \mathbf{B}} \|\mathbf{X}^T\mathbf{X} - \mathbf{B}\|_F$

Example: Word Embedding

LSA gives a way of embedding words into k-dimensional space.

- Embedding is via low-rank approximation of $\mathbf{X}^T\mathbf{X}$: where $(\mathbf{X}^T\mathbf{X})_{a,b}$ is the number of documents that both $word_a$ and $word_b$ appear in.

Example: Word Embedding

LSA gives a way of embedding words into k-dimensional space.

- Embedding is via low-rank approximation of $\mathbf{X}^T\mathbf{X}$: where $(\mathbf{X}^T\mathbf{X})_{a,b}$ is the number of documents that both $word_a$ and $word_b$ appear in.
- Think about $\mathbf{X}^T\mathbf{X}$ as a similarity matrix (gram matrix, kernel matrix) with entry (a, b) being the similarity between $word_a$ and $word_b$.

Example: Word Embedding

LSA gives a way of embedding words into k-dimensional space.

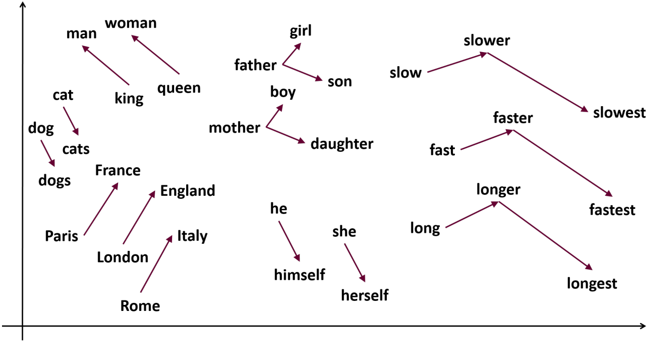
- Embedding is via low-rank approximation of $\mathbf{X}^T\mathbf{X}$: where $(\mathbf{X}^T\mathbf{X})_{a,b}$ is the number of documents that both $word_a$ and $word_b$ appear in.
- Think about $\mathbf{X}^T\mathbf{X}$ as a similarity matrix (gram matrix, kernel matrix) with entry (a, b) being the similarity between $word_a$ and $word_b$.
- Many ways to measure similarity: number of sentences both occur in, number of times both appear in the same window of w words, in similar positions of documents in different languages, etc.

Example: Word Embedding

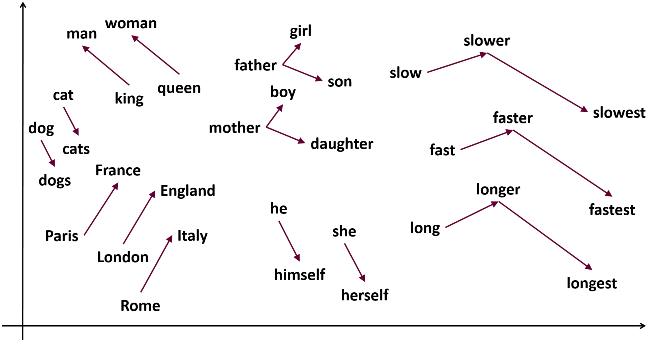
LSA gives a way of embedding words into k-dimensional space.

- Embedding is via low-rank approximation of $X^T X$: where $(X^T X)_{a,b}$ is the number of documents that both $word_a$ and $word_b$ appear in.
- Think about $X^T X$ as a similarity matrix (gram matrix, kernel matrix) with entry (a, b) being the similarity between $word_a$ and $word_b$.
- Many ways to measure similarity: number of sentences both occur in, number of times both appear in the same window of w words, in similar positions of documents in different languages, etc.
- Replacing $X^T X$ with these different metrics (sometimes appropriately transformed) leads to popular word embedding algorithms: word2vec, GloVe, fastText, etc.

Example: Word Embedding



Example: Word Embedding



Note: word2vec is typically described as a neural-network method, but can be viewed as just a low-rank approximation of a specific similarity matrix. *Neural word embedding as implicit matrix factorization*, Levy and Goldberg.