COMS 4771 Lecture 19

1. Principal component analysis
2. Singular value decomposition

Representation learning

Useful representations of data

Representation learning:

- **Given**: raw feature vectors $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)} \in \mathbb{R}^d$.
- **Goal**: learn a "useful" feature transformation $\boldsymbol{\phi}\colon \mathbb{R}^d \to \mathbb{R}^k$.
  (Often $k \ll d$—i.e., *dimensionality reduction*—but not always.)

Useful representations of data

Representation learning:

- **Given**: raw feature vectors $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)} \in \mathbb{R}^d$.
- **Goal**: learn a "useful" feature transformation $\boldsymbol{\phi}\colon \mathbb{R}^d \to \mathbb{R}^k$.
  (Often $k \ll d$—i.e., *dimensionality reduction*—but not always.)

  Can then use $\boldsymbol{\phi}$ as a feature map for supervised learning.

Useful representations of data

Representation learning:

- **Given**: raw feature vectors $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)} \in \mathbb{R}^d$.
- **Goal**: learn a "useful" feature transformation $\boldsymbol{\phi}\colon \mathbb{R}^d \to \mathbb{R}^k$.
  (Often $k \ll d$—i.e., *dimensionality reduction*—but not always.)

  Can then use $\boldsymbol{\phi}$ as a feature map for supervised learning.

Some previously encontered examples:

- Feature maps corresponding to pos. def. kernels (+approximations).
  (Usually *data-oblivious*—feature map doesn't depend on the data.)
- Centering $\boldsymbol{x} \mapsto \boldsymbol{x} - \boldsymbol{\mu}$
  (Effect: all features have mean $\mathbf{0}$.)
- Standardization $\boldsymbol{x} \mapsto \mathrm{Diag}(\sigma_1, \sigma_2, \ldots, \sigma_d)^{-1}(\boldsymbol{x} - \boldsymbol{\mu})$.
  (Effect: all features have mean $\mathbf{0}$ and unit variance.)

Useful representations of data

Representation learning:

- **Given**: raw feature vectors $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)} \in \mathbb{R}^d$.
- **Goal**: learn a "useful" feature transformation $\boldsymbol{\phi}\colon \mathbb{R}^d \to \mathbb{R}^k$.
  (Often $k \ll d$—i.e., *dimensionality reduction*—but not always.)

  Can then use $\boldsymbol{\phi}$ as a feature map for supervised learning.

Some previously encontered examples:

- Feature maps corresponding to pos. def. kernels (+approximations).
  (Usually *data-oblivious*—feature map doesn't depend on the data.)
- Centering $\boldsymbol{x} \mapsto \boldsymbol{x} - \boldsymbol{\mu}$
  (Effect: all features have mean $\mathbf{0}$.)
- Standardization $\boldsymbol{x} \mapsto \mathrm{Diag}(\sigma_1, \sigma_2, \ldots, \sigma_d)^{-1}(\boldsymbol{x} - \boldsymbol{\mu})$.
  (Effect: all features have mean $\mathbf{0}$ and unit variance.)

What other properties of a feature representation may be desirable?

Principal component analysis

Projections

- **Input**: $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(n)} \in \mathbb{R}^d$, target dimensionality $k \in \mathbb{N}$.
- **Output**: a k-dimensional subspace, represented by an orthonormal basis $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_k \in \mathbb{R}^d$.
- **Projection**: Formally, projection of $\boldsymbol{x} \in \mathbb{R}^d$ to $\operatorname{span}(\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_k)$ is

$$\underbrace{\left(\sum_{i=1}^{k} \boldsymbol{q}_i \boldsymbol{q}_i^\top\right)}_{\boldsymbol{\Pi}} \boldsymbol{x} \;=\; \sum_{i=1}^{k} \langle \boldsymbol{q}_i, \boldsymbol{x} \rangle \boldsymbol{q}_i \;\in\; \mathbb{R}^d.$$

But, we can simply represent the projection in terms of its coefficients w.r.t. the orthonormal basis $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_k$:

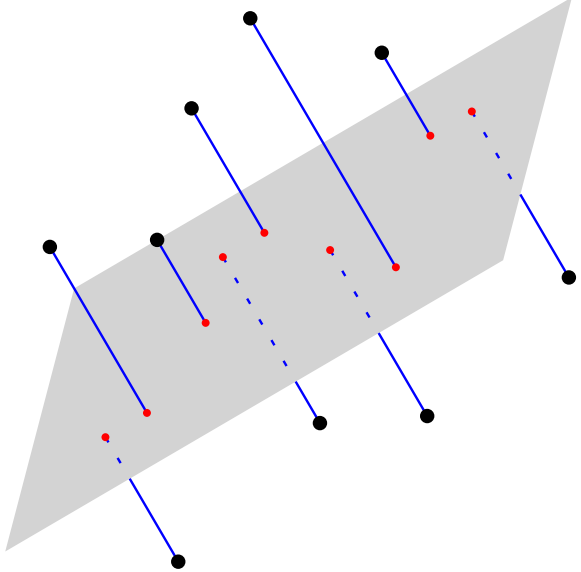
$$\boldsymbol{\phi}(\boldsymbol{x}) \;:=\; \begin{bmatrix} \langle \boldsymbol{q}_1, \boldsymbol{x} \rangle \\ \langle \boldsymbol{q}_2, \boldsymbol{x} \rangle \\ \vdots \\ \langle \boldsymbol{q}_k, \boldsymbol{x} \rangle \end{bmatrix} \;\in\; \mathbb{R}^k.$$

Minimize residual squared error

Objective: find k-dimensional projector $\mathbf{\Pi}\colon \mathbb{R}^d \to \mathbb{R}^d$ such that the average residual squared error

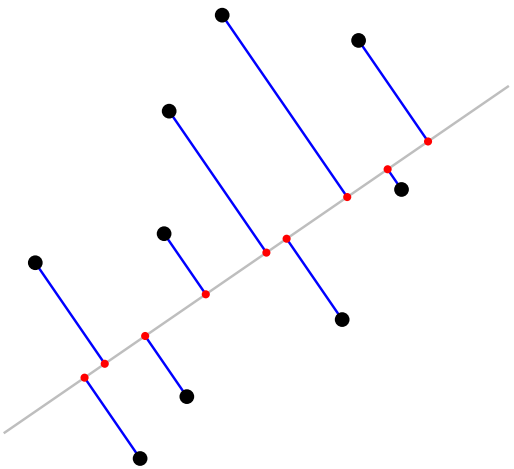
$$\frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{x}^{(i)} - \mathbf{\Pi}\boldsymbol{x}^{(i)}\right\|_2^2$$

is as small as possible.



Projection of minimum residual squared error

$k = 1$ **case** ($\boldsymbol{\Pi} = \boldsymbol{q}\boldsymbol{q}^\top$)

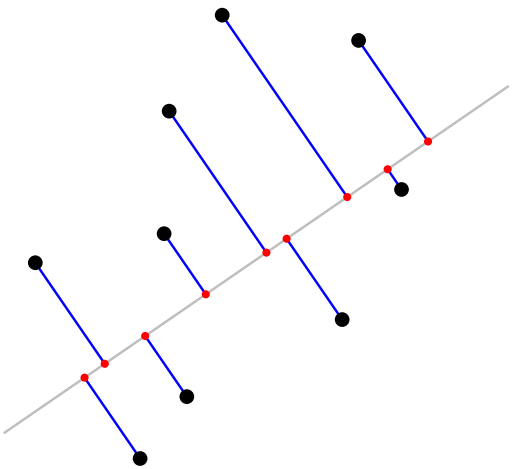


Objective: find unit vector $\boldsymbol{q} \in \mathbb{R}^d$ to minimize

$$\frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{x}^{(i)} - \boldsymbol{q}\boldsymbol{q}^\top\boldsymbol{x}^{(i)}\right\|_2^2$$

Projection of minimum residual squared error

$k = 1$ **case** ($\boldsymbol{\Pi} = \boldsymbol{q}\boldsymbol{q}^\top$)



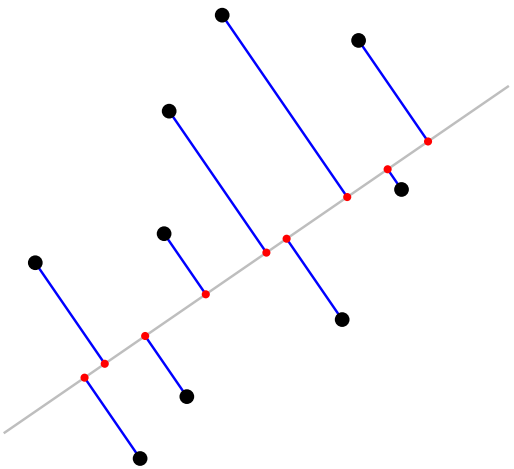
Objective: find unit vector $\boldsymbol{q} \in \mathbb{R}^d$ to minimize

$$\frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{x}^{(i)} - \boldsymbol{q}\boldsymbol{q}^\top\boldsymbol{x}^{(i)}\right\|_2^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{x}^{(i)}\|_2^2 - \boldsymbol{q}^\top\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}^{(i)}\boldsymbol{x}^{(i)\top}\right)\boldsymbol{q}$$

Projection of minimum residual squared error

$k = 1$ **case ($\boldsymbol{\Pi} = \boldsymbol{q}\boldsymbol{q}^\top$)**



Objective: find unit vector $\boldsymbol{q} \in \mathbb{R}^d$ to minimize

$$\frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{x}^{(i)} - \boldsymbol{q}\boldsymbol{q}^\top\boldsymbol{x}^{(i)}\right\|_2^2$$

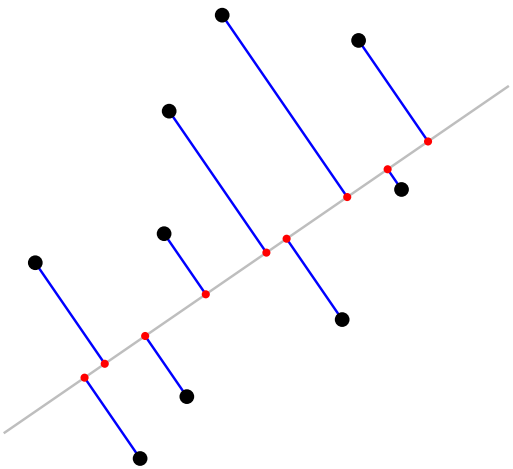
$$= \frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{x}^{(i)}\|_2^2 - \boldsymbol{q}^\top\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}^{(i)}\boldsymbol{x}^{(i)\top}\right)\boldsymbol{q}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{x}^{(i)}\|_2^2 - \boldsymbol{q}^\top\left(\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}\right)\boldsymbol{q}$$

(where $\boldsymbol{x}^{(i)\top}$ is i-th row of $\boldsymbol{X} \in \mathbb{R}^{n\times d}$).

Projection of minimum residual squared error

$k = 1$ **case ($\boldsymbol{\Pi} = \boldsymbol{q}\boldsymbol{q}^\top$)**



Objective: find unit vector $\boldsymbol{q} \in \mathbb{R}^d$ to minimize

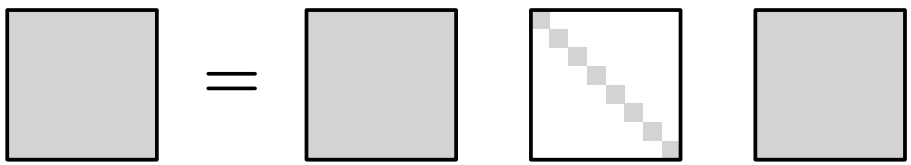
$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{x}^{(i)} - \boldsymbol{q}\boldsymbol{q}^\top\boldsymbol{x}^{(i)}\right\|_2^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{x}^{(i)}\|_2^2 - \boldsymbol{q}^\top\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}^{(i)}\boldsymbol{x}^{(i)\top}\right)\boldsymbol{q} \\
&= \frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{x}^{(i)}\|_2^2 - \boldsymbol{q}^\top\left(\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}\right)\boldsymbol{q}
\end{aligned}$$

(where $\boldsymbol{x}^{(i)\top}$ is i-th row of $\boldsymbol{X} \in \mathbb{R}^{n\times d}$).

$$\underset{\boldsymbol{q}\in\mathbb{R}^d:\|\boldsymbol{q}\|_2=1}{\arg\min}\ \frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{x}^{(i)} - \boldsymbol{q}\boldsymbol{q}^\top\boldsymbol{x}^{(i)}\right\|_2^2 \quad\equiv\quad \underset{\boldsymbol{q}\in\mathbb{R}^d:\|\boldsymbol{q}\|_2=1}{\arg\max}\ \boldsymbol{q}^\top\left(\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}\right)\boldsymbol{q}.$$

Eigendecompositions

Every symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ guaranteed to have eigendecomposition with real eigenvalues:



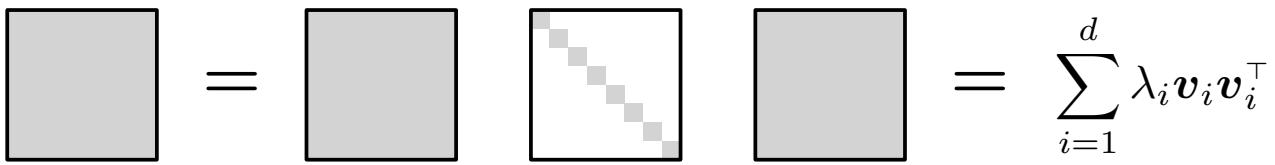
$$= \sum_{i=1}^{d} \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^\top$$

$\boldsymbol{A}$ $(d \times d)$ $\quad$ $\boldsymbol{V}$ $(d \times d)$ $\quad$ $\boldsymbol{\Lambda}$ $(d \times d)$ $\quad$ $\boldsymbol{V}^\top$ $(d \times d)$

real **eigenvalues**: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \quad (\boldsymbol{\Lambda} = \mathrm{Diag}(\lambda_1, \lambda_2, \ldots, \lambda_d))$;
corresponding orthonormal **eigenvectors**: $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_d \quad (\boldsymbol{V} = [\boldsymbol{v}_1 | \boldsymbol{v}_2 | \cdots | \boldsymbol{v}_d])$.

Eigendecompositions

Every symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ guaranteed to have eigendecomposition with real eigenvalues:



$$\boldsymbol{A} = \boldsymbol{V} \boldsymbol{\Lambda} \boldsymbol{V}^\top = \sum_{i=1}^{d} \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^\top$$

$\boldsymbol{A}$ $(d \times d)$ $\quad$ $\boldsymbol{V}$ $(d \times d)$ $\quad$ $\boldsymbol{\Lambda}$ $(d \times d)$ $\quad$ $\boldsymbol{V}^\top$ $(d \times d)$

real **eigenvalues**: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \quad (\boldsymbol{\Lambda} = \mathrm{Diag}(\lambda_1, \lambda_2, \ldots, \lambda_d))$;
corresponding orthonormal **eigenvectors**: $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_d \quad (\boldsymbol{V} = [\boldsymbol{v}_1 | \boldsymbol{v}_2 | \cdots | \boldsymbol{v}_d])$.

Fixed-point characterization of eigenvectors:

$$\boldsymbol{A}\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i.$$

Variational characterization of eigenvectors:

$$\max_{\boldsymbol{q} \in \mathbb{R}^d} \boldsymbol{q}^\top \boldsymbol{A} \boldsymbol{q}$$
$$\text{s.t. } \|\boldsymbol{q}\|_2 = 1$$

- Maximum value: λ_1 (top eigenvalue)
- Maximizer: $\boldsymbol{v}_1$ (top eigenvector)

Variational characterization of eigenvectors:

$$\max_{\boldsymbol{q} \in \mathbb{R}^d} \boldsymbol{q}^\top \boldsymbol{A} \boldsymbol{q}$$
$$\text{s.t. } \|\boldsymbol{q}\|_2 = 1$$

- Maximum value: λ_1 (top eigenvalue)
- Maximizer: $\boldsymbol{v}_1$ (top eigenvector)

For $i > 1$,

$$\max_{\boldsymbol{q} \in \mathbb{R}^d} \boldsymbol{q}^\top \boldsymbol{A} \boldsymbol{q}$$
$$\text{s.t. } \|\boldsymbol{q}\|_2 = 1$$
$$\langle \boldsymbol{q}, \boldsymbol{v}_j \rangle = 0 \ \forall j < i$$

- Maximum value: λ_i (i-th largest eigenvalue)
- Maximizer: $\boldsymbol{v}_i$ (i-th eigenvector)

Principal component analysis ($k = 1$)

$k = 1$ **case ($\boldsymbol{\Pi} = \boldsymbol{q}\boldsymbol{q}^\top$)**

$$\underset{\boldsymbol{q}\in\mathbb{R}^d:\|\boldsymbol{q}\|_2=1}{\arg\min} \frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{x}^{(i)} - \boldsymbol{q}\boldsymbol{q}^\top\boldsymbol{x}^{(i)}\right\|_2^2 \quad \equiv \quad \underset{\boldsymbol{q}\in\mathbb{R}^d:\|\boldsymbol{q}\|_2=1}{\arg\max} \boldsymbol{q}^\top\left(\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}\right)\boldsymbol{q}.$$

Solution: eigenvector of $\boldsymbol{X}^\top\boldsymbol{X}$ corresponding to largest eigenvalue (i.e., the top eigenvector $\boldsymbol{v}_1$).

Principal component analysis ($k = 1$)

$k = 1$ **case ($\boldsymbol{\Pi} = \boldsymbol{q}\boldsymbol{q}^\top$)**

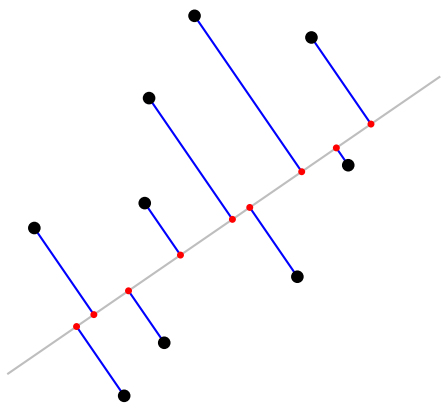
$$\underset{\boldsymbol{q}\in\mathbb{R}^d:\|\boldsymbol{q}\|_2=1}{\arg\min} \frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{x}^{(i)} - \boldsymbol{q}\boldsymbol{q}^\top\boldsymbol{x}^{(i)}\right\|_2^2 \quad \equiv \quad \underset{\boldsymbol{q}\in\mathbb{R}^d:\|\boldsymbol{q}\|_2=1}{\arg\max} \boldsymbol{q}^\top\left(\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}\right)\boldsymbol{q}.$$

Solution: eigenvector of $\boldsymbol{X}^\top\boldsymbol{X}$ corresponding to largest eigenvalue (i.e., the top eigenvector $\boldsymbol{v}_1$).

$$\boldsymbol{q}^\top\left(\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}\right)\boldsymbol{q} \;=\; \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{q}^\top\boldsymbol{x}^{(i)})^2 \;=\; \text{empirical variance of } \boldsymbol{q}^\top\boldsymbol{x}$$

(assuming $\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}^{(i)} = \mathbf{0}$—i.e., "centering" already applied).

top eigenvector $\equiv$ direction of maximum variance



Principal component analysis (general k)

General k case ($\boldsymbol{\Pi} = \boldsymbol{Q}\boldsymbol{Q}^\top$)

$$\underset{\substack{\boldsymbol{Q}\in\mathbb{R}^{d\times k}:\\ \boldsymbol{Q}^\top\boldsymbol{Q}=\boldsymbol{I}}}{\arg\min}\ \frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{x}^{(i)} - \boldsymbol{Q}\boldsymbol{Q}^\top\boldsymbol{x}^{(i)}\right\|_2^2 \quad \equiv \quad \underset{\substack{\boldsymbol{Q}\in\mathbb{R}^{d\times k}:\\ \boldsymbol{Q}^\top\boldsymbol{Q}=\boldsymbol{I}}}{\arg\max}\ \operatorname{tr}\left(\boldsymbol{Q}^\top\left(\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}\right)\boldsymbol{Q}\right).$$

Solution: k eigenvectors of $\boldsymbol{X}^\top\boldsymbol{X}$ corresponding to k largest eigenvalue

Principal component analysis (general k)

General k case ($\boldsymbol{\Pi} = \boldsymbol{Q}\boldsymbol{Q}^\top$)

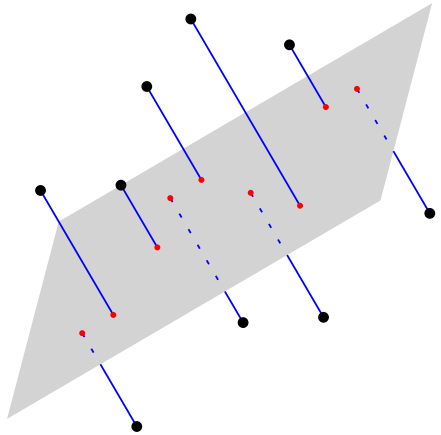
$$\underset{\substack{\boldsymbol{Q}\in\mathbb{R}^{d\times k}:\\ \boldsymbol{Q}^\top\boldsymbol{Q}=\boldsymbol{I}}}{\arg\min}\ \frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{x}^{(i)} - \boldsymbol{Q}\boldsymbol{Q}^\top\boldsymbol{x}^{(i)}\right\|_2^2 \quad\equiv\quad \underset{\substack{\boldsymbol{Q}\in\mathbb{R}^{d\times k}:\\ \boldsymbol{Q}^\top\boldsymbol{Q}=\boldsymbol{I}}}{\arg\max}\ \mathrm{tr}\left(\boldsymbol{Q}^\top\left(\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}\right)\boldsymbol{Q}\right).$$

Solution: k eigenvectors of $\boldsymbol{X}^\top\boldsymbol{X}$ corresponding to k largest eigenvalue

$$\mathrm{tr}\left(\boldsymbol{Q}^\top\left(\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}\right)\boldsymbol{Q}\right) = \sum_{i=1}^{k}\text{empirical variance of } \boldsymbol{q}_i^\top\boldsymbol{x}$$

(assuming $\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}^{(i)} = \mathbf{0}$—i.e., "centering" already applied).

top k eigenvectors $\equiv$ k-dim. subspace of maximum variance



Principal component analysis (PCA)

Data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$

Rank k PCA (k dimensional linear subspace)

- Get top k eigenvectors $\boldsymbol{V} = [\boldsymbol{v}_1 | \boldsymbol{v}_2 | \dots | \boldsymbol{v}_k]$ of

$$\frac{1}{n} \boldsymbol{X}^\top \boldsymbol{X} \;=\; \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}^{(i)} \boldsymbol{x}^{(i)\top}.$$

- *Feature map*: $\boldsymbol{\phi}(\boldsymbol{x}) := (\langle \boldsymbol{v}_1, \boldsymbol{x} \rangle, \langle \boldsymbol{v}_2, \boldsymbol{x} \rangle, \dots, \langle \boldsymbol{v}_k, \boldsymbol{x} \rangle) \in \mathbb{R}^k$
- *Decorrelating property*:

$$\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\phi}(\boldsymbol{x}^{(i)}) \boldsymbol{\phi}(\boldsymbol{x}^{(i)})^\top = \mathrm{Diag}(\lambda_1, \lambda_2, \dots, \lambda_k)$$

- *Reconstruction*: $\boldsymbol{x} \mapsto \boldsymbol{V} \boldsymbol{\phi}(\boldsymbol{x})$

Principal component analysis (PCA)

Data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$

Rank k PCA with centering (k dimensional affine subspace)

- Get top k eigenvectors $\boldsymbol{V} = [\boldsymbol{v}_1 | \boldsymbol{v}_2 | \dots | \boldsymbol{v}_k]$ of

$$\frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu})(\boldsymbol{x}^{(i)} - \boldsymbol{\mu})^\top$$

  where $\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}^{(i)}$.

- *Feature map*: $\boldsymbol{\phi}(\boldsymbol{x}) := (\langle \boldsymbol{v}_1, \boldsymbol{x} - \boldsymbol{\mu} \rangle, \langle \boldsymbol{v}_2, \boldsymbol{x} - \boldsymbol{\mu} \rangle, \dots, \langle \boldsymbol{v}_k, \boldsymbol{x} - \boldsymbol{\mu} \rangle) \in \mathbb{R}^k$

- *Decorrelating property*:

$$\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\phi}(\boldsymbol{x}^{(i)}) = \boldsymbol{0}$$

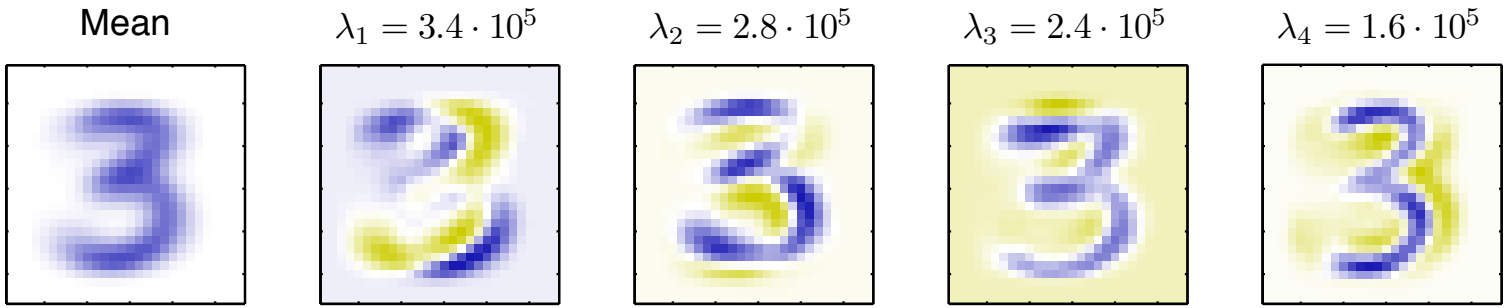
$$\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\phi}(\boldsymbol{x}^{(i)}) \boldsymbol{\phi}(\boldsymbol{x}^{(i)})^\top = \mathrm{Diag}(\lambda_1, \lambda_2, \dots, \lambda_k)$$

- *Reconstruction*: $\boldsymbol{x} \mapsto \boldsymbol{\mu} + \boldsymbol{V} \boldsymbol{\phi}(\boldsymbol{x})$

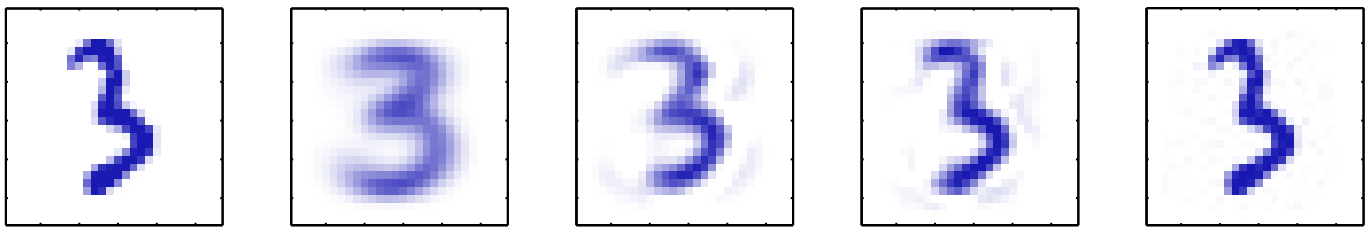
Example: compressing digits images

16×16 pixel images of handwritten 3s (as vectors in $\mathbb{R}^{256}$)

Mean $\boldsymbol{\mu}$ and eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4$



Reconstructions:



Only have to store k numbers per image,
along with the mean $\boldsymbol{\mu}$ and k eigenvectors ($256(k+1)$ numbers).

Example: eigenfaces

92×112 pixel images of faces (as vectors in $\mathbb{R}^{10304}$)

100 example images

top $k = 48$ eigenvectors

Other examples

- $\boldsymbol{x} \in \mathbb{R}^d$: movement of stock prices for d different stocks in one day.

Other examples

- $\boldsymbol{x} \in \mathbb{R}^d$: movement of stock prices for d different stocks in one day.

  Principal component: combination of stocks that account for the most variation in stock price movement.

Other examples

- $\boldsymbol{x} \in \mathbb{R}^d$: movement of stock prices for d different stocks in one day.

  Principal component: combination of stocks that account for the most variation in stock price movement.

- $\boldsymbol{x} \in \{1, 2, \ldots, 5\}^d$: levels at which various terms describe an individual (e.g., "jolly", "impulsive", "outgoing", "conceited", "meddlesome")

Other examples

- $\boldsymbol{x} \in \mathbb{R}^d$: movement of stock prices for d different stocks in one day.

  Principal component: combination of stocks that account for the most variation in stock price movement.

- $\boldsymbol{x} \in \{1, 2, \ldots, 5\}^d$: levels at which various terms describe an individual (e.g., "jolly", "impulsive", "outgoing", "conceited", "meddlesome")

  Principal components: major personality axes in a population (e.g., "extroversion", "agreeableness", "conscientiousness")

Other examples

- $\boldsymbol{x} \in \mathbb{R}^d$: movement of stock prices for d different stocks in one day.

  Principal component: combination of stocks that account for the most variation in stock price movement.

- $\boldsymbol{x} \in \{1, 2, \ldots, 5\}^d$: levels at which various terms describe an individual (e.g., "jolly", "impulsive", "outgoing", "conceited", "meddlesome")

  Principal components: major personality axes in a population (e.g., "extroversion", "agreeableness", "conscientiousness")

- ...

Computation

Power method

Problem: Given matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, compute the top eigenvector of $\boldsymbol{X}^\top \boldsymbol{X}$.

- **Initialize** with random $\hat{\boldsymbol{v}} \in \mathbb{R}^d$.
- **Repeat**:
  1. $\hat{\boldsymbol{v}} := \boldsymbol{X}^\top \boldsymbol{X} \hat{\boldsymbol{v}}$.
  2. $\hat{\boldsymbol{v}} := \hat{\boldsymbol{v}} / \|\hat{\boldsymbol{v}}\|_2$.

Power method

Problem: Given matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, compute the top eigenvector of $\boldsymbol{X}^\top \boldsymbol{X}$.

- **Initialize** with random $\hat{\boldsymbol{v}} \in \mathbb{R}^d$.
- **Repeat**:
  1. $\hat{\boldsymbol{v}} := \boldsymbol{X}^\top \boldsymbol{X} \hat{\boldsymbol{v}}$.
  2. $\hat{\boldsymbol{v}} := \hat{\boldsymbol{v}} / \|\hat{\boldsymbol{v}}\|_2$.

Theorem: For any $\varepsilon \in (0, 1)$, with high probability (over choice of initial $\hat{\boldsymbol{v}}$),

$$\hat{\boldsymbol{v}}^\top \left(\frac{1}{n} \boldsymbol{X}^\top \boldsymbol{X} \right) \hat{\boldsymbol{v}} \;\geq\; (1 - \varepsilon) \cdot \text{top eigenvalue of } \frac{1}{n} \boldsymbol{X}^\top \boldsymbol{X}$$

after $O\left(\frac{1}{\varepsilon} \log \frac{d}{\varepsilon} \right)$ iterations.

Power method

Problem: Given matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, compute the top eigenvector of $\boldsymbol{X}^\top \boldsymbol{X}$.

- **Initialize** with random $\hat{\boldsymbol{v}} \in \mathbb{R}^d$.
- **Repeat**:
  1. $\hat{\boldsymbol{v}} := \boldsymbol{X}^\top \boldsymbol{X} \hat{\boldsymbol{v}}$.
  2. $\hat{\boldsymbol{v}} := \hat{\boldsymbol{v}} / \|\hat{\boldsymbol{v}}\|_2$.

Theorem: For any $\varepsilon \in (0, 1)$, with high probability (over choice of initial $\hat{\boldsymbol{v}}$),

$$\hat{\boldsymbol{v}}^\top \left(\frac{1}{n} \boldsymbol{X}^\top \boldsymbol{X} \right) \hat{\boldsymbol{v}} \;\geq\; (1 - \varepsilon) \cdot \text{top eigenvalue of } \frac{1}{n} \boldsymbol{X}^\top \boldsymbol{X}$$

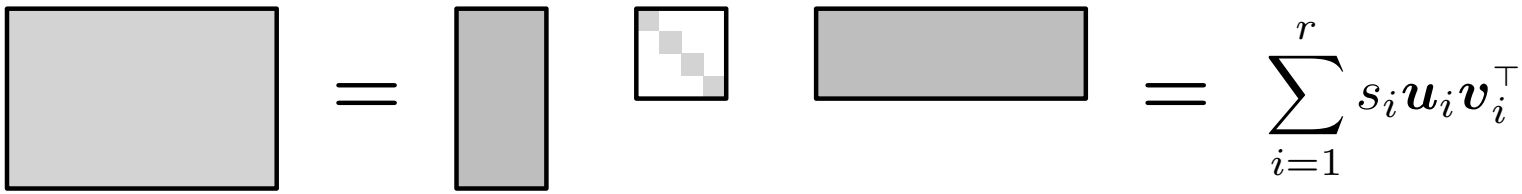
after $O\left(\frac{1}{\varepsilon} \log \frac{d}{\varepsilon} \right)$ iterations.

Similar algorithm can be used to get top k eigenvectors.

Singular value decomposition

Singular value decomposition

Every matrix $\boldsymbol{A} \in \mathbb{R}^{n \times d}$ has a **singular value decomposition (SVD)**



$$= \sum_{i=1}^{r} s_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$$

$\boldsymbol{A}$ $(n \times d)$ $\quad$ $\boldsymbol{U}$ $(n \times r)$ $\quad$ $\boldsymbol{S}$ $(r \times r)$ $\quad$ $\boldsymbol{V}^\top$ $(r \times d)$

where

- $r = \operatorname{rank}(\boldsymbol{A})$ $\quad$ $(r \leq \min\{n, d\})$;
- $\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}$ $\quad$ (i.e., $\boldsymbol{U} = [\boldsymbol{u}_1 | \boldsymbol{u}_2 | \cdots | \boldsymbol{u}_r]$ has orthonormal columns)

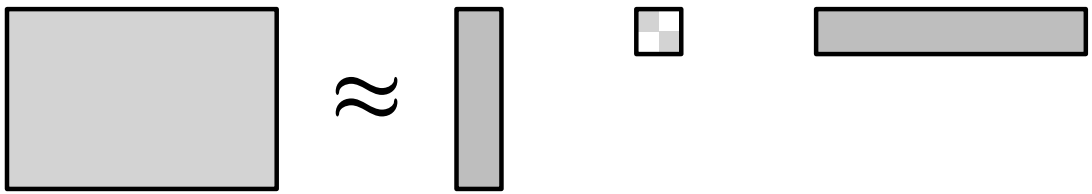
  left singular vectors;
- $\boldsymbol{S} = \operatorname{Diag}(s_1, s_2, \ldots, s_r)$ $\quad$ where $s_1 \geq s_2 \geq \cdots \geq s_r > 0$

  singular values;
- $\boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{I}$ $\quad$ (i.e., $\boldsymbol{V} = [\boldsymbol{v}_1 | \boldsymbol{v}_2 | \cdots | \boldsymbol{v}_r]$ has orthonormal columns)

  right singular vectors.

Low-rank SVD

For any $k \leq \text{rank}(\boldsymbol{A})$, **rank-$k$ SVD approximation**:



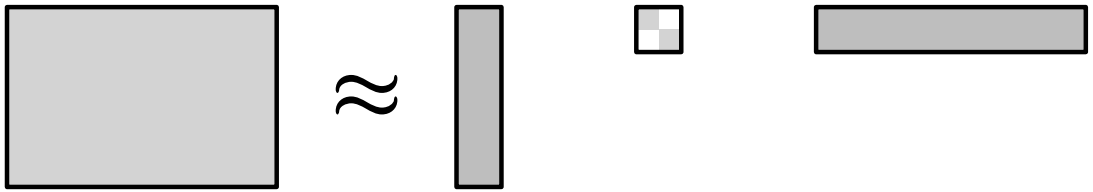
$$\approx \quad \widehat{\boldsymbol{U}}_k \quad \widehat{\boldsymbol{S}}_k \quad \widehat{\boldsymbol{V}}_k^\top \quad = \quad \sum_{i=1}^{k} s_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$$

$\boldsymbol{A}$ $(n \times d)$ $\quad$ $\widehat{\boldsymbol{U}}_k$ $(n \times k)$ $\quad$ $\widehat{\boldsymbol{S}}_k$ $(k \times k)$ $\quad$ $\widehat{\boldsymbol{V}}_k^\top$ $(k \times d)$

(Just retain top k left/right singular vectors and singular values from SVD.)

Low-rank SVD

For any $k \leq \text{rank}(\boldsymbol{A})$, **rank-$k$ SVD approximation**:



$$= \sum_{i=1}^{k} s_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$$

$$\underset{(n \times d)}{\boldsymbol{A}} \approx \underset{(n \times k)}{\widehat{\boldsymbol{U}}_k} \; \underset{(k \times k)}{\widehat{\boldsymbol{S}}_k} \; \underset{(k \times d)}{\widehat{\boldsymbol{V}}_k^\top}$$

(Just retain top k left/right singular vectors and singular values from SVD.)

Best rank-k approximation:

$$\widehat{\boldsymbol{A}} := \widehat{\boldsymbol{U}}_k \widehat{\boldsymbol{S}}_k \widehat{\boldsymbol{V}}_k^\top = \underset{\substack{\boldsymbol{M} \in \mathbb{R}^{n \times d}: \\ \text{rank}(\boldsymbol{M}) \leq k}}{\arg\min} \sum_{i=1}^{n} \sum_{j=1}^{d} (A_{i,j} - M_{i,j})^2.$$

Minimum value is simply given by

$$\sum_{i=1}^{n} \sum_{j=1}^{d} (A_{i,j} - \widehat{A}_{i,j})^2 = \sum_{t>k} s_t^2.$$

Example: latent semantic analysis

Represent corpus of documents by counts of words they contain:

	aardvark	abacus	abalone	$\cdots$
document 1	3	0	0	$\cdots$
document 2	7	0	4	$\cdots$
document 3	2	4	0	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	

- One column per vocabulary word in $\boldsymbol{A} \in \mathbb{R}^{n \times d}$
- One row per document in $\boldsymbol{A} \in \mathbb{R}^{n \times d}$
- $A_{i,j} =$ numbers of times word j appears in document i.

Example: latent semantic analysis

Modeling assumption:

- $k \ll \min\{n, d\}$ "topics", each represented by a distributions over vocabulary words:

  $$\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_k \in \mathbb{R}^d$$

  (Each $\boldsymbol{\beta}_t = (\beta_{t,1}, \beta_{t,2}, \ldots, \beta_{t,d})$ is a probability vector.)

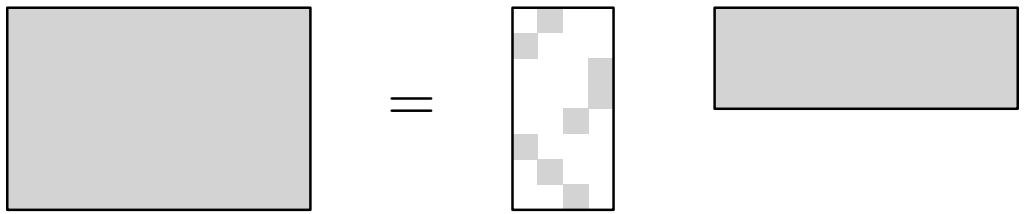
- Each document i is associated with a topic $t_i \in \{1, 2, \ldots, k\}$.

  Document i's count vector (i-th row in $\boldsymbol{A}$) is drawn from a multinomial distribution with probabilities given by $\boldsymbol{\beta}_{t_i}$.

Example: latent semantic analysis

Implication of modeling assumption

In expectation, $\boldsymbol{A}$ has rank k:



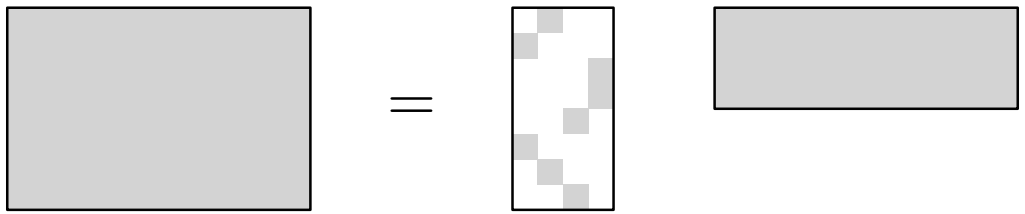
$\mathbb{E}(\boldsymbol{A})$ $(n \times d)$ $\quad =$ $\boldsymbol{L}$ $(n \times k)$ $\quad$ $\boldsymbol{B}^\top$ $(k \times d)$

- L_{i,t_i} = length of document i (other entries are zero).
- $\boldsymbol{\beta}_t = t$-th column of $\boldsymbol{B}$

Example: latent semantic analysis

Implication of modeling assumption

In expectation, $\boldsymbol{A}$ has rank k:



$\mathbb{E}(\boldsymbol{A})$ $(n \times d)$ $\quad=\quad$ $\boldsymbol{L}$ $(n \times k)$ $\quad$ $\boldsymbol{B}^\top$ $(k \times d)$

- L_{i,t_i} = length of document i (other entries are zero).
- $\boldsymbol{\beta}_t = t$-th column of $\boldsymbol{B}$

Observed matrix $\boldsymbol{A}$:

$$\boldsymbol{A} \;=\; \mathbb{E}(\boldsymbol{A}) + \textbf{Zero mean noise}$$

so $\boldsymbol{A}$ is generally of rank $\min\{n, d\} \gg k$.

Example: latent semantic analysis

Using SVD: rank-k SVD $\widehat{\boldsymbol{U}}_k \widehat{\boldsymbol{S}}_k \widehat{\boldsymbol{V}}_k^\top$ of $\boldsymbol{A}$ gives approximation to $\boldsymbol{L}\boldsymbol{B}^\top$:

$$\widehat{\boldsymbol{A}} := \widehat{\boldsymbol{U}}_k \widehat{\boldsymbol{S}}_k \widehat{\boldsymbol{V}}_k^\top \approx \mathbb{E}(\boldsymbol{A}) = \boldsymbol{L}\boldsymbol{B}^\top.$$

(SVD helps remove some of the effect of the noise.)

Example: latent semantic analysis

Using SVD: rank-k SVD $\widehat{\boldsymbol{U}}_k \widehat{\boldsymbol{S}}_k \widehat{\boldsymbol{V}}_k^\top$ of $\boldsymbol{A}$ gives approximation to $\boldsymbol{L}\boldsymbol{B}^\top$:

$$\widehat{\boldsymbol{A}} := \widehat{\boldsymbol{U}}_k \widehat{\boldsymbol{S}}_k \widehat{\boldsymbol{V}}_k^\top \approx \mathbb{E}(\boldsymbol{A}) = \boldsymbol{L}\boldsymbol{B}^\top.$$

(SVD helps remove some of the effect of the noise.)

- Each of the n documents can be summarized by k numbers:

$$\widehat{\boldsymbol{A}}\widehat{\boldsymbol{V}}_k = \widehat{\boldsymbol{U}}_k \widehat{\boldsymbol{S}}_k \in \mathbb{R}^{n \times k}.$$

Example: latent semantic analysis

Using SVD: rank-k SVD $\widehat{\boldsymbol{U}}_k \widehat{\boldsymbol{S}}_k \widehat{\boldsymbol{V}}_k^\top$ of $\boldsymbol{A}$ gives approximation to $\boldsymbol{L}\boldsymbol{B}^\top$:

$$\widehat{\boldsymbol{A}} := \widehat{\boldsymbol{U}}_k \widehat{\boldsymbol{S}}_k \widehat{\boldsymbol{V}}_k^\top \approx \mathbb{E}(\boldsymbol{A}) = \boldsymbol{L}\boldsymbol{B}^\top.$$

(SVD helps remove some of the effect of the noise.)

- Each of the n documents can be summarized by k numbers:

$$\widehat{\boldsymbol{A}}\widehat{\boldsymbol{V}}_k = \widehat{\boldsymbol{U}}_k \widehat{\boldsymbol{S}}_k \in \mathbb{R}^{n \times k}.$$

- New document representation very useful for information retrieval.

  (Example: cosine similarities between documents become faster to compute and possibly less noisy.)

Example: latent semantic analysis

Using SVD: rank-k SVD $\widehat{\boldsymbol{U}}_k \widehat{\boldsymbol{S}}_k \widehat{\boldsymbol{V}}_k^\top$ of $\boldsymbol{A}$ gives approximation to $\boldsymbol{L}\boldsymbol{B}^\top$:

$$\widehat{\boldsymbol{A}} := \widehat{\boldsymbol{U}}_k \widehat{\boldsymbol{S}}_k \widehat{\boldsymbol{V}}_k^\top \approx \mathbb{E}(\boldsymbol{A}) = \boldsymbol{L}\boldsymbol{B}^\top.$$

(SVD helps remove some of the effect of the noise.)

- Each of the n documents can be summarized by k numbers:

$$\widehat{\boldsymbol{A}}\widehat{\boldsymbol{V}}_k = \widehat{\boldsymbol{U}}_k \widehat{\boldsymbol{S}}_k \in \mathbb{R}^{n \times k}.$$

- New document representation very useful for information retrieval.

  (Example: cosine similarities between documents become faster to compute and possibly less noisy.)

- Actually estimating $\boldsymbol{L}$ and $\boldsymbol{B}$ takes a bit more work.

Connection to PCA

- Let $\boldsymbol{X}$ be the design matrix. Suppose we want to compute the rank k PCA for $\boldsymbol{X}$.

Connection to PCA

- Let $\boldsymbol{X}$ be the design matrix. Suppose we want to compute the rank k PCA for $\boldsymbol{X}$.
- Let $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$ be the SVD of $\boldsymbol{X}$.

Connection to PCA

- Let $\boldsymbol{X}$ be the design matrix. Suppose we want to compute the rank k PCA for $\boldsymbol{X}$.
- Let $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$ be the SVD of $\boldsymbol{X}$.
- PCA: Compute top k eigenvalues/eigenvectors of $\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}$:

$$\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X} = \frac{1}{n}\boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^\top\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top = \frac{1}{n}\boldsymbol{V}\boldsymbol{S}^2\boldsymbol{V}^\top.$$

Connection to PCA

- Let $\boldsymbol{X}$ be the design matrix. Suppose we want to compute the rank k PCA for $\boldsymbol{X}$.
- Let $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$ be the SVD of $\boldsymbol{X}$.
- PCA: Compute top k eigenvalues/eigenvectors of $\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}$:

$$\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X} = \frac{1}{n}\boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^\top\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top = \frac{1}{n}\boldsymbol{V}\boldsymbol{S}^2\boldsymbol{V}^\top.$$

- Lo and behold - the SVD yields the eigendecomposition of $\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}$!
- Eigenvalues are $\frac{1}{n}$ times the diagonal entries in $\boldsymbol{S}^2$, and eigenvectors are columns of $\boldsymbol{V}$.

Connection to PCA

- Let $\boldsymbol{X}$ be the design matrix. Suppose we want to compute the rank k PCA for $\boldsymbol{X}$.
- Let $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$ be the SVD of $\boldsymbol{X}$.
- PCA: Compute top k eigenvalues/eigenvectors of $\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}$:

$$\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X} = \frac{1}{n}\boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^\top\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top = \frac{1}{n}\boldsymbol{V}\boldsymbol{S}^2\boldsymbol{V}^\top.$$

- Lo and behold - the SVD yields the eigendecomposition of $\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}$!
- Eigenvalues are $\frac{1}{n}$ times the diagonal entries in $\boldsymbol{S}^2$, and eigenvectors are columns of $\boldsymbol{V}$.
- Even better: PCA embedding of a point $\boldsymbol{\phi}(\boldsymbol{x}) = (\langle \boldsymbol{v}_1, \boldsymbol{x}\rangle, \langle \boldsymbol{v}_2, \boldsymbol{x}\rangle, \ldots, \langle \boldsymbol{v}_k, \boldsymbol{x}\rangle)^\top$, is simply $\boldsymbol{V}_k^\top\boldsymbol{x}$.
- So PCA embedding of *all* training points are the columns of $\boldsymbol{V}_k^\top\boldsymbol{X}^\top$.

Connection to PCA

- Let $\boldsymbol{X}$ be the design matrix. Suppose we want to compute the rank k PCA for $\boldsymbol{X}$.
- Let $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$ be the SVD of $\boldsymbol{X}$.
- PCA: Compute top k eigenvalues/eigenvectors of $\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}$:

$$\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X} = \frac{1}{n}\boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^\top\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top = \frac{1}{n}\boldsymbol{V}\boldsymbol{S}^2\boldsymbol{V}^\top.$$

- Lo and behold - the SVD yields the eigendecomposition of $\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}$!
- Eigenvalues are $\frac{1}{n}$ times the diagonal entries in $\boldsymbol{S}^2$, and eigenvectors are columns of $\boldsymbol{V}$.
- Even better: PCA embedding of a point $\boldsymbol{\phi}(\boldsymbol{x}) = (\langle \boldsymbol{v}_1, \boldsymbol{x}\rangle, \langle \boldsymbol{v}_2, \boldsymbol{x}\rangle, \ldots, \langle \boldsymbol{v}_k, \boldsymbol{x}\rangle)^\top$, is simply $\boldsymbol{V}_k^\top\boldsymbol{x}$.
- So PCA embedding of *all* training points are the columns of $\boldsymbol{V}_k^\top\boldsymbol{X}^\top$.
- But $\boldsymbol{V}_k^\top = \boldsymbol{S}_k^{-1}\boldsymbol{U}_k^\top\boldsymbol{X}$, so we can also write the PCA embedding as $\boldsymbol{S}_k^{-1}\boldsymbol{U}_k^\top\boldsymbol{X}\boldsymbol{X}^\top$. Useful for kernelization (hw!)

Recap

- **PCA**: directions of maximum variance in data $\equiv$ subspace that minimizes residual squared error.
- **Computation**: power method
- **SVD**: general decomposition for arbitrary rectangular matrices

  Low-rank SVD: best low-rank approximation of a matrix
- **PCA/SVD**: often useful when low-rank structure is expected (e.g., probabilistic modeling).