The Singular Value Decomposition

DS-GA 1013 / MATH-GA 2824 Mathematical Tools for Data Science

https://cims.nyu.edu/~cfgranda/pages/MTDS_spring19/index.html

Carlos Fernandez-Granda

Motivating applications

The singular-value decomposition

Principal component analysis

Dimensionality reduction

Linear regression

Ridge regression

Low-rank matrix estimation

Dimensionality reduction

Data with a large number of features can be difficult to analyze

Data modeled as vectors in $\mathbb{R}^p$ (p very large)

Aim: Reduce dimensionality of representation

Intuition: Directions with very little variation are useless

Problem: How to find directions of maximum/minimum variation

Regression

The aim is to learn a function h that relates

- a response or dependent variable y
- to several observed variables $\vec{x} \in \mathbb{R}^p$, known as covariates, features or independent variables

The response is assumed to be of the form

$$y \approx h(\vec{x})$$

Linear regression

The regression function h is assumed to be linear

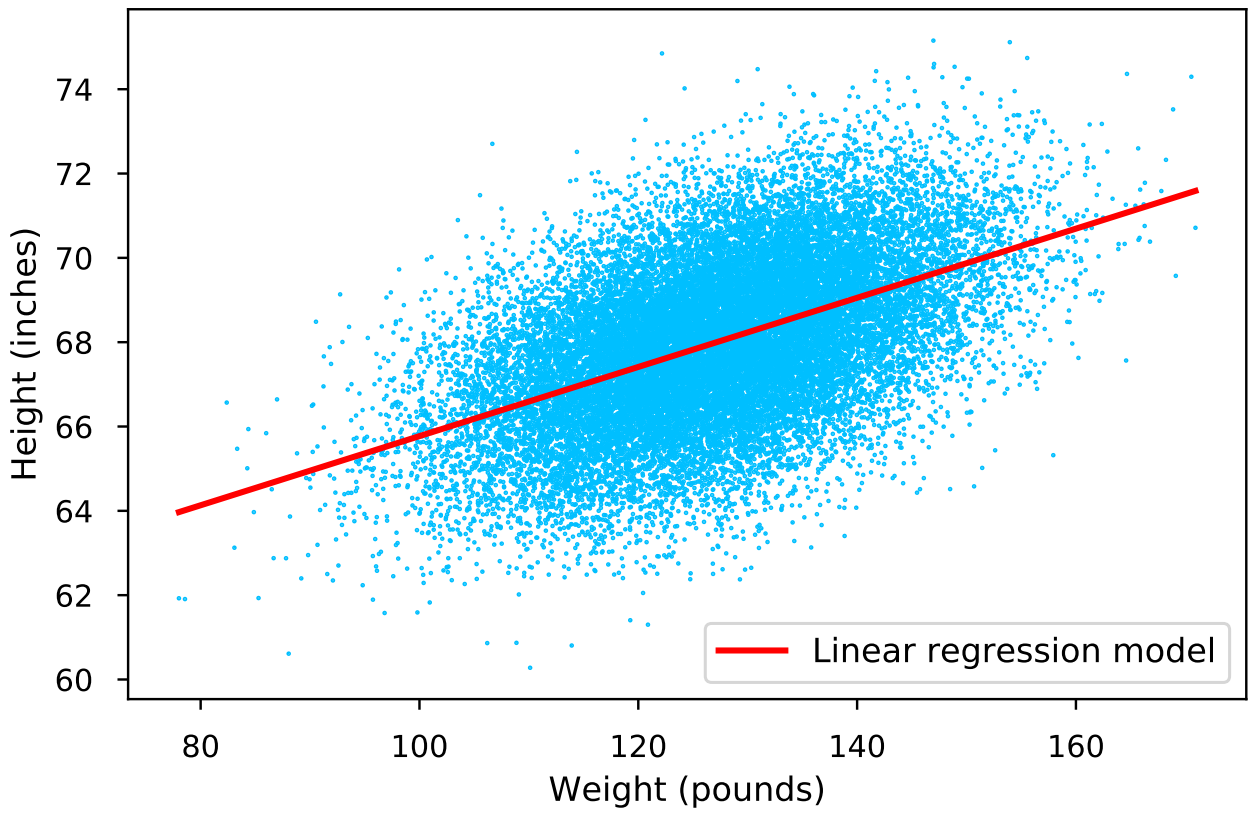
$$y^{(i)} \approx \vec{x}^{(i)\,T} \vec{\beta} + \beta_0, \quad 1 \leq i \leq n$$

We estimate $\vec{\beta} \in \mathbb{R}^p$ from training dataset

$$\mathcal{S}_{\text{train}} := \left\{ \left(y^{(1)}, \vec{x}^{(1)} \right), \left(y^{(2)}, \vec{x}^{(2)} \right), \ldots, \left(y^{(n)}, \vec{x}^{(n)} \right) \right\}$$

A crucial question is how well we do on held-out test data

Linear regression



Collaborative filtering

Quantity $y[i, j]$ depends on indices i and j

We observe examples and want to predict new instances

For example, $y[i, j]$ is rating given to a movie i by a user j

Collaborative filtering

Figure courtesy of Mahdi Soltanolkotabi

Rank-1 bilinear model

- Some movies are more popular in general
- Some users are more generous in general

$$y[i,j] \approx a[i]b[j]$$

- $a[i]$ quantifies popularity of movie i
- $b[j]$ quantifies generosity of user j

Rank-r bilinear model

Certain people like certain movies: r factors

$$y[i,j] \approx \sum_{l=1}^{r} a_l[i] b_l[j]$$

For each factor l

- $a_l[i]$: movie i is positively (> 0), negatively (< 0) or not (≈ 0) associated to factor l
- $b_l[j]$: user j is positively (> 0), negatively (< 0) or not (≈ 0) associated to factor l

Singular-value decomposition can be used to fit the model

Motivating applications

The singular-value decomposition

Principal component analysis

Dimensionality reduction

Linear regression

Ridge regression

Low-rank matrix estimation

Singular value decomposition

Every rank r real matrix $A \in R^{m \times n}$, has a singular-value decomposition (SVD) of the form

$$A = \begin{bmatrix} \vec{u}_1 & \vec{u}_2 & \cdots & \vec{u}_r \end{bmatrix} \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \cdots & s_r \end{bmatrix} \begin{bmatrix} \vec{v}_1^T \\ \vec{v}_2^T \\ \vdots \\ \vec{v}_r^T \end{bmatrix}$$

$$= USV^T$$

Singular value decomposition

- The singular values $s_1 \geq s_2 \geq \cdots \geq s_r$ are positive real numbers
- The left singular vectors $\vec{u}_1$, $\vec{u}_2$, $\ldots \vec{u}_r$ form an orthonormal set
- The right singular vectors $\vec{v}_1$, $\vec{v}_2$, $\ldots \vec{v}_r$ also form an orthonormal set
- The SVD is unique if all the singular values are different
- If $s_i = s_{i+1} = \ldots = s_{i+k}$, then $\vec{u}_i$, $\ldots$, $\vec{u}_{i+k}$ can be replaced by any orthonormal basis of their span (the same holds for $\vec{v}_i$, $\ldots$, $\vec{v}_{i+k}$)
- The SVD of an $m \times n$ matrix with $m \geq n$ can be computed in $\mathcal{O}\left(mn^2\right)$

Column and row space

- The left singular vectors $\vec{u}_1, \vec{u}_2, \ldots \vec{u}_r$ are a basis for the column space
- The right singular vectors $\vec{v}_1, \vec{v}_2, \ldots \vec{v}_r$ are a basis for the row space

Proof:

$\text{span}\,(\vec{u}_1, \ldots, \vec{u}_r) \subseteq \text{col}\,(A)$

$\text{col}\,(A) \subseteq \text{span}\,(\vec{u}_1, \ldots, \vec{u}_r)$

Column and row space

- The left singular vectors $\vec{u}_1, \vec{u}_2, \ldots \vec{u}_r$ are a basis for the column space
- The right singular vectors $\vec{v}_1, \vec{v}_2, \ldots \vec{v}_r$ are a basis for the row space

Proof:

$\text{span}\left(\vec{u}_1, \ldots, \vec{u}_r\right) \subseteq \text{col}\left(A\right)$ because $\vec{u}_i = A\left(s_i^{-1}\vec{v}_i\right)$

$\text{col}\left(A\right) \subseteq \text{span}\left(\vec{u}_1, \ldots, \vec{u}_r\right)$

Column and row space

- The left singular vectors $\vec{u}_1, \vec{u}_2, \ldots \vec{u}_r$ are a basis for the column space
- The right singular vectors $\vec{v}_1, \vec{v}_2, \ldots \vec{v}_r$ are a basis for the row space

Proof:

$\text{span}\left(\vec{u}_1, \ldots, \vec{u}_r\right) \subseteq \text{col}\left(A\right)$ because $\vec{u}_i = A\left(s_i^{-1}\vec{v}_i\right)$

$\text{col}\left(A\right) \subseteq \text{span}\left(\vec{u}_1, \ldots, \vec{u}_r\right)$ because $A_{:i} = U\left(SV^T\vec{e}_i\right)$

Singular value decomposition

$$A = [\underbrace{\vec{u}_1 \cdots \vec{u}_r}_{\text{Basis of range}(A)} \quad \vec{u}_{r+1} \cdots \vec{u}_n] \begin{bmatrix} s_1 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & s_r & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ & & \cdots & \cdots & & \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix} [\underbrace{\vec{v}_1 \, \vec{v}_2 \cdots \vec{v}_r}_{\text{Basis of row}(A)} \quad \underbrace{\vec{v}_{r+1} \cdots \vec{v}_n}_{\text{Basis of null}(A)}]^T$$

Linear maps

The SVD decomposes the action of a matrix $A \in \mathbb{R}^{m \times n}$ on a vector $\vec{x} \in \mathbb{R}^n$ into:

1. Rotation

$$V^T \vec{x} = \sum_{i=1}^{n} \langle \vec{v}_i, \vec{x} \rangle \, \vec{e}_i$$

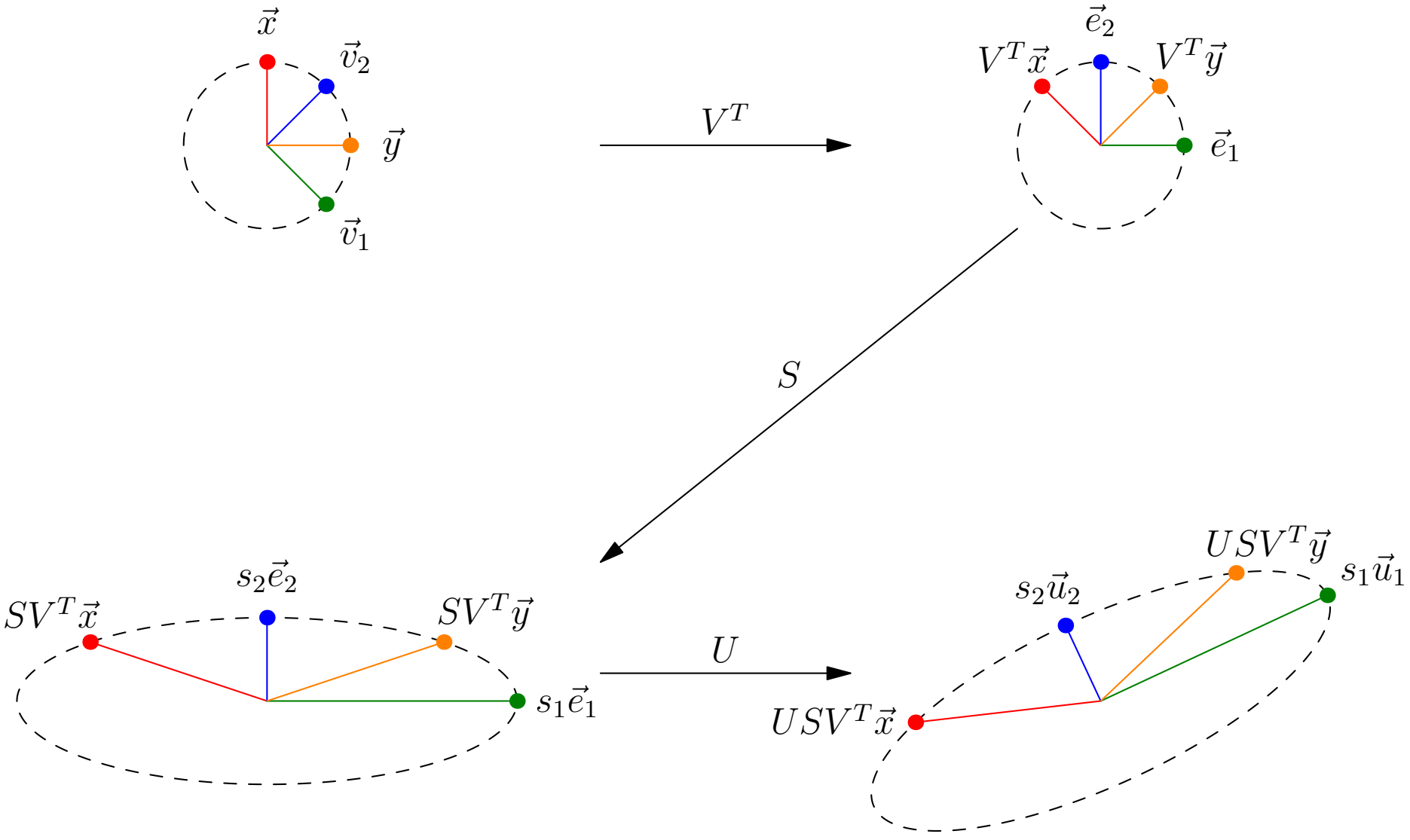
2. Scaling

$$SV^T \vec{x} = \sum_{i=1}^{n} s_i \, \langle \vec{v}_i, \vec{x} \rangle \, \vec{e}_i$$

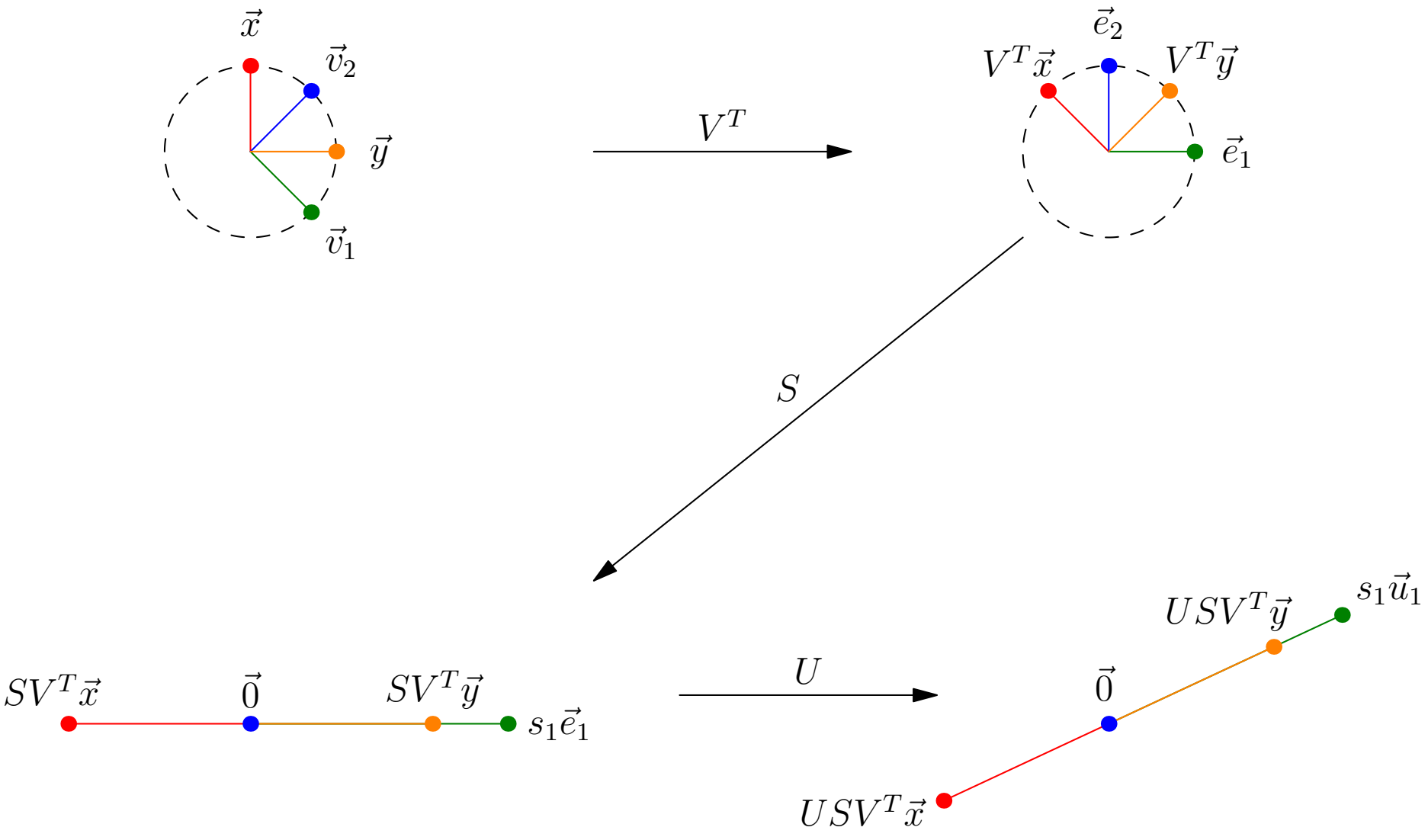
3. Rotation

$$USV^T \vec{x} = \sum_{i=1}^{n} s_i \, \langle \vec{v}_i, \vec{x} \rangle \, \vec{u}_i$$

Linear maps


$\vec{x}$
$\vec{v}_2$
$\vec{y}$
$\vec{v}_1$
V^T
$\vec{e}_2$
$V^T\vec{x}$
$V^T\vec{y}$
$\vec{e}_1$
S
$s_2\vec{e}_2$
$SV^T\vec{x}$
$SV^T\vec{y}$
$s_1\vec{e}_1$
U
$USV^T\vec{y}$
$s_2\vec{u}_2$
$s_1\vec{u}_1$
$USV^T\vec{x}$

Linear maps ($s_2 := 0$)



Singular values

The singular values satisfy

$$
\begin{aligned}
s_1 &= \max_{\left\{||\vec{x}||_2=1 \mid \vec{x}\in\mathbb{R}^n\right\}} ||A\vec{x}||_2 \\
&= \max_{\left\{||\vec{y}||_2=1 \mid \vec{y}\in\mathbb{R}^m\right\}} \left|\left|A^T\vec{y}\right|\right|_2 \\
s_i &= \max_{\left\{||\vec{x}||_2=1 \mid \vec{x}\in\mathbb{R}^n, \vec{x}\perp\vec{u}_1,\ldots,\vec{u}_{i-1}\right\}} ||A\vec{x}||_2 \\
&= \max_{\left\{||\vec{y}||_2=1 \mid \vec{y}\in\mathbb{R}^m, \vec{y}\perp\vec{v}_1,\ldots,\vec{v}_{i-1}\right\}} \left|\left|A^T\vec{y}\right|\right|_2, \qquad 2 \leq i \leq \min\left\{m,n\right\}
\end{aligned}
$$

Singular vectors

The right singular vectors satisfy

$$\vec{v}_1 = \underset{\left\{||\vec{x}||_2=1 \mid \vec{x}\in\mathbb{R}^n\right\}}{\arg\max} ||A\vec{x}||_2$$

$$\vec{v}_i = \underset{\left\{||\vec{x}||_2=1 \mid \vec{x}\in\mathbb{R}^n, \vec{x}\perp\vec{v}_1,\ldots,\vec{v}_{i-1}\right\}}{\arg\max} ||A\vec{x}||_2, \qquad 2 \leq i \leq m$$

and the left singular vectors satisfy

$$\vec{u}_1 = \underset{\left\{||\vec{y}||_2=1 \mid \vec{y}\in\mathbb{R}^m\right\}}{\arg\max} \left|\left|A^T\vec{y}\right|\right|_2$$

$$\vec{u}_i = \underset{\left\{||\vec{y}||_2=1 \mid \vec{y}\in\mathbb{R}^m, \vec{y}\perp\vec{u}_1,\ldots,\vec{u}_{i-1}\right\}}{\arg\max} \left|\left|A^T\vec{y}\right|\right|_2, \qquad 2 \leq i \leq n$$

Proof

$$||A\vec{v}_i||_2^2$$

Proof

$$||A\vec{v}_i||_2^2 = \left\langle \sum_{k=1}^{n} s_k \left\langle \vec{v}_k, \vec{v}_i \right\rangle \vec{u}_k, \sum_{k=1}^{n} s_k \left\langle \vec{v}_k, \vec{v}_i \right\rangle \vec{u}_k \right\rangle$$

Proof

$$
\begin{aligned}
||A\vec{v}_i||_2^2 &= \left\langle \sum_{k=1}^{n} s_k \left\langle \vec{v}_k, \vec{v}_i \right\rangle \vec{u}_k, \sum_{k=1}^{n} s_k \left\langle \vec{v}_k, \vec{v}_i \right\rangle \vec{u}_k \right\rangle \\
&= \sum_{k=1}^{n} s_k^2 \left\langle \vec{v}_k, \vec{v}_i \right\rangle^2
\end{aligned}
$$

Proof

$$
\begin{aligned}
||A\vec{v}_i||_2^2 &= \left\langle \sum_{k=1}^{n} s_k \left\langle \vec{v}_k, \vec{v}_i \right\rangle \vec{u}_k, \sum_{k=1}^{n} s_k \left\langle \vec{v}_k, \vec{v}_i \right\rangle \vec{u}_k \right\rangle \\
&= \sum_{k=1}^{n} s_k^2 \left\langle \vec{v}_k, \vec{v}_i \right\rangle^2 \\
&= s_i^2
\end{aligned}
$$

Proof

Still need to prove that no other vector achieves a larger value

Proof

Still need to prove that no other vector achieves a larger value

Consider $\vec{x} \in \mathbb{R}^n$ such that $||\vec{x}||_2 = 1$ and for a fixed $1 \leq i \leq n$

$$\vec{x} \perp \vec{v}_1, \ldots, \vec{v}_{i-1}$$

We decompose $\vec{x}$ into

$$\vec{x} = \sum_{j=i}^{n} \alpha_j \vec{v}_j + \mathcal{P}_{\text{row}(A)^\perp} \vec{x}$$

where $1 = ||\vec{x}||_2^2 \geq \sum_{j=i}^{n} \alpha_j^2$

Proof

$$||A\vec{x}||_2^2$$

Proof

$$||A\vec{x}||_2^2 = \left\langle \sum_{k=1}^{n} s_k \left\langle \vec{v}_k, \vec{x} \right\rangle \vec{u}_k, \sum_{k=1}^{n} s_k \left\langle \vec{v}_k, \vec{x} \right\rangle \vec{u}_k \right\rangle$$

Proof

$$
\begin{aligned}
||A\vec{x}||_2^2 &= \left\langle \sum_{k=1}^{n} s_k \left\langle \vec{v}_k, \vec{x} \right\rangle \vec{u}_k, \sum_{k=1}^{n} s_k \left\langle \vec{v}_k, \vec{x} \right\rangle \vec{u}_k \right\rangle \\
&= \sum_{k=1}^{n} s_k^2 \left\langle \vec{v}_k, \vec{x} \right\rangle^2 \quad \text{because } \vec{u}_1, \ldots, \vec{u}_n \text{ are orthonormal}
\end{aligned}
$$

Proof

$$
\begin{aligned}
||A\vec{x}||_2^2 &= \left\langle \sum_{k=1}^{n} s_k \left\langle \vec{v}_k, \vec{x} \right\rangle \vec{u}_k, \sum_{k=1}^{n} s_k \left\langle \vec{v}_k, \vec{x} \right\rangle \vec{u}_k \right\rangle \\
&= \sum_{k=1}^{n} s_k^2 \left\langle \vec{v}_k, \vec{x} \right\rangle^2 \quad \text{because } \vec{u}_1, \ldots, \vec{u}_n \text{ are orthonormal} \\
&= \sum_{k=1}^{n} s_k^2 \left\langle \vec{v}_k, \sum_{j=i}^{n} \alpha_j \vec{v}_j + \mathcal{P}_{\text{row}(A)^\perp} \vec{x} \right\rangle^2
\end{aligned}
$$

Proof

$$
\begin{aligned}
||A\vec{x}||_2^2 &= \left\langle \sum_{k=1}^{n} s_k \left\langle \vec{v}_k, \vec{x} \right\rangle \vec{u}_k, \sum_{k=1}^{n} s_k \left\langle \vec{v}_k, \vec{x} \right\rangle \vec{u}_k \right\rangle \\
&= \sum_{k=1}^{n} s_k^2 \left\langle \vec{v}_k, \vec{x} \right\rangle^2 \quad \text{because } \vec{u}_1, \ldots, \vec{u}_n \text{ are orthonormal} \\
&= \sum_{k=1}^{n} s_k^2 \left\langle \vec{v}_k, \sum_{j=i}^{n} \alpha_j \vec{v}_j + \mathcal{P}_{\text{row}(A)^\perp} \vec{x} \right\rangle^2 \\
&= \sum_{j=i}^{n} s_j^2 \alpha_j^2 \quad \text{because } \vec{v}_1, \ldots, \vec{v}_n \text{ are orthonormal}
\end{aligned}
$$

Proof

$$
\begin{aligned}
||A\vec{x}||_2^2 &= \left\langle \sum_{k=1}^{n} s_k \left\langle \vec{v}_k, \vec{x} \right\rangle \vec{u}_k, \sum_{k=1}^{n} s_k \left\langle \vec{v}_k, \vec{x} \right\rangle \vec{u}_k \right\rangle \\
&= \sum_{k=1}^{n} s_k^2 \left\langle \vec{v}_k, \vec{x} \right\rangle^2 \quad \text{because } \vec{u}_1, \ldots, \vec{u}_n \text{ are orthonormal} \\
&= \sum_{k=1}^{n} s_k^2 \left\langle \vec{v}_k, \sum_{j=i}^{n} \alpha_j \vec{v}_j + \mathcal{P}_{\text{row}(A)^\perp} \vec{x} \right\rangle^2 \\
&= \sum_{j=i}^{n} s_j^2 \alpha_j^2 \quad \text{because } \vec{v}_1, \ldots, \vec{v}_n \text{ are orthonormal} \\
&\leq s_i^2 \sum_{j=i}^{n} \alpha_j^2 \quad \text{because } s_i \geq s_{i+1} \geq \ldots \geq s_n
\end{aligned}
$$

Proof

$$
\begin{aligned}
||A\vec{x}||_2^2 &= \left\langle \sum_{k=1}^{n} s_k \left\langle \vec{v}_k, \vec{x} \right\rangle \vec{u}_k, \sum_{k=1}^{n} s_k \left\langle \vec{v}_k, \vec{x} \right\rangle \vec{u}_k \right\rangle \\
&= \sum_{k=1}^{n} s_k^2 \left\langle \vec{v}_k, \vec{x} \right\rangle^2 \quad \text{because } \vec{u}_1, \ldots, \vec{u}_n \text{ are orthonormal} \\
&= \sum_{k=1}^{n} s_k^2 \left\langle \vec{v}_k, \sum_{j=i}^{n} \alpha_j \vec{v}_j + \mathcal{P}_{\text{row}(A)^\perp} \vec{x} \right\rangle^2 \\
&= \sum_{j=i}^{n} s_j^2 \alpha_j^2 \quad \text{because } \vec{v}_1, \ldots, \vec{v}_n \text{ are orthonormal} \\
&\leq s_i^2 \sum_{j=i}^{n} \alpha_j^2 \quad \text{because } s_i \geq s_{i+1} \geq \ldots \geq s_n \\
&\leq s_i^2
\end{aligned}
$$

Motivating applications

The singular-value decomposition

Principal component analysis

Dimensionality reduction

Linear regression

Ridge regression

Low-rank matrix estimation

Quantifying directional variation

Goal: Quantify variation of a dataset embedded in a p-dimensional space

Two possible perspectives:

- **Probabilistic**: The data are samples from a p-dimensional random vector $\vec{\mathbf{x}}$
- **Geometric**: The data are just a cloud of points in $\mathbb{R}^p$

Probabilistic perspective ($p = 1$)

A natural quantifier of variation is the variance of the distribution

$$\mathrm{Var}\left(\mathbf{x}\right) := \mathrm{E}\left(\left(\mathbf{x} - \mathrm{E}\left(\mathbf{x}\right)\right)^2\right)$$

Geometric perspective ($p = 1$)

A natural quantifier of variation is the sample variance of the data

$$\text{var}\left(x_1, x_2, \ldots, x_n\right) := \frac{1}{n-1} \sum_{i=1}^{n} \left(x_i - \text{av}\left(x_1, x_2, \ldots, x_n\right)\right)^2$$

$$\text{av}\left(x_1, x_2, \ldots, x_n\right) := \frac{1}{n} \sum_{i=1}^{n} x_i$$

Connection

Sample variance is an estimator of variance

If $\{\mathsf{x}_1, \mathsf{x}_2, \ldots, \mathsf{x}_n\}$ have mean μ and variance σ^2,

$$\mathrm{E}\left(\mathsf{av}\left(\mathsf{x}_1, \mathsf{x}_2, \ldots, \mathsf{x}_n\right)\right) = \mu$$
$$\mathrm{E}\left(\mathsf{var}\left(\mathsf{x}_1, \mathsf{x}_2, \ldots, \mathsf{x}_n\right)\right) = \sigma^2$$

by linearity of expectation, so the estimator is unbiased

Connection

If samples are independent, again by linearity of expectation,

$$\mathrm{E}\left((\mathsf{av}\,(\mathsf{x}_1, \mathsf{x}_2, \ldots, \mathsf{x}_n) - \mu)^2\right) = \frac{\sigma^2}{n}$$

Connection

If samples are independent, again by linearity of expectation,

$$\mathrm{E}\left(\left(\mathrm{av}\left(\mathsf{x}_1, \mathsf{x}_2, \ldots, \mathsf{x}_n\right) - \mu\right)^2\right) = \frac{\sigma^2}{n}$$

If fourth moment is bounded,

$$\mathrm{E}\left(\left(\mathrm{var}\left(\mathsf{x}_1, \mathsf{x}_2, \ldots, \mathsf{x}_n\right) - \sigma^2\right)^2\right)$$

also scales as $1/n$

Probabilistic perspective ($p > 1$)

1. Center data by subtracting mean

2. Variation in direction $\vec{v}$ can be quantified by $\mathrm{Var}\left(\vec{v}^{\,T}\vec{\mathbf{x}}\right)$

Geometric perspective ($p > 1$)

1. Center data by subtracting average

2. Variation in direction $\vec{v}$ quantified by $\text{var}\left(\vec{v}^T\vec{x}_1, \vec{v}^T\vec{x}_2, \ldots, \vec{v}^T\vec{x}_n\right)$

Covariance

The covariance of two random variables $\mathbf{x}$ and $\mathbf{y}$ is

$$\operatorname{Cov}(\mathbf{x}, \mathbf{y}) := \operatorname{E}((\mathbf{x} - \operatorname{E}(\mathbf{x}))(\mathbf{y} - \operatorname{E}(\mathbf{y})))$$

Covariance matrix

The covariance matrix of a random vector $\vec{\mathbf{x}}$ is defined as

$$\Sigma_{\vec{\mathbf{x}}} := \begin{bmatrix} \operatorname{Var}(\vec{\mathbf{x}}[1]) & \operatorname{Cov}(\vec{\mathbf{x}}[1], \vec{\mathbf{x}}[2]) & \cdots & \operatorname{Cov}(\vec{\mathbf{x}}[1], \vec{\mathbf{x}}[p]) \\ \operatorname{Cov}(\vec{\mathbf{x}}[2], \vec{\mathbf{x}}[1]) & \operatorname{Var}(\vec{\mathbf{x}}[2]) & \cdots & \operatorname{Cov}(\vec{\mathbf{x}}[2], \vec{\mathbf{x}}[p]) \\ \vdots & \vdots & \ddots & \vdots \\ \operatorname{Cov}(\vec{\mathbf{x}}[n], \vec{\mathbf{x}}[1]) & \operatorname{Cov}(\vec{\mathbf{x}}[n], \vec{\mathbf{x}}[2]) & \cdots & \operatorname{Var}(\vec{\mathbf{x}}[p]) \end{bmatrix}$$

$$= \mathrm{E}(\vec{\mathbf{x}}\vec{\mathbf{x}}^T) - \mathrm{E}(\vec{\mathbf{x}})\mathrm{E}(\vec{\mathbf{x}})^T.$$

If the covariance matrix is diagonal, the entries are uncorrelated

Covariance matrix after a linear transformation

For any matrix $A \in \mathbb{R}^{m \times n}$ and n-dimensional random vector $\vec{\mathbf{x}}$, the covariance matrix of $A\vec{\mathbf{x}}$ equals

$$\Sigma_{A\vec{\mathbf{x}}} = A\Sigma_{\vec{\mathbf{x}}}A^T$$

Covariance matrix after a linear transformation

For any matrix $A \in \mathbb{R}^{m \times n}$ and n-dimensional random vector $\vec{\mathbf{x}}$, the covariance matrix of $A\vec{\mathbf{x}}$ equals

$$\Sigma_{A\vec{\mathbf{x}}} = A\Sigma_{\vec{\mathbf{x}}}A^T$$

Proof:

$$\Sigma_{A\vec{\mathbf{x}}}$$

Covariance matrix after a linear transformation

For any matrix $A \in \mathbb{R}^{m \times n}$ and n-dimensional random vector $\vec{\mathbf{x}}$, the covariance matrix of $A\vec{\mathbf{x}}$ equals

$$\Sigma_{A\vec{\mathbf{x}}} = A\Sigma_{\vec{\mathbf{x}}}A^T$$

Proof:

$$\Sigma_{A\vec{\mathbf{x}}} = \mathrm{E}\left(\left(A\vec{\mathbf{x}}\right)\left(A\vec{\mathbf{x}}\right)^T\right) - \mathrm{E}\left(A\vec{\mathbf{x}}\right)\mathrm{E}\left(A\vec{\mathbf{x}}\right)^T$$

Covariance matrix after a linear transformation

For any matrix $A \in \mathbb{R}^{m \times n}$ and n-dimensional random vector $\vec{\mathbf{x}}$, the covariance matrix of $A\vec{\mathbf{x}}$ equals

$$\Sigma_{A\vec{\mathbf{x}}} = A \Sigma_{\vec{\mathbf{x}}} A^T$$

Proof:

$$\begin{aligned} \Sigma_{A\vec{\mathbf{x}}} &= \mathrm{E}\left((A\vec{\mathbf{x}}) \, (A\vec{\mathbf{x}})^T \right) - \mathrm{E}\left(A\vec{\mathbf{x}}\right) \mathrm{E}\left(A\vec{\mathbf{x}}\right)^T \\ &= A \left(\mathrm{E}\left(\vec{\mathbf{x}} \vec{\mathbf{x}}^T \right) - \mathrm{E}(\vec{\mathbf{x}}) \mathrm{E}(\vec{\mathbf{x}})^T \right) A^T \end{aligned}$$

Covariance matrix after a linear transformation

For any matrix $A \in \mathbb{R}^{m \times n}$ and n-dimensional random vector $\vec{\mathbf{x}}$, the covariance matrix of $A\vec{\mathbf{x}}$ equals

$$\Sigma_{A\vec{\mathbf{x}}} = A\Sigma_{\vec{\mathbf{x}}}A^T$$

Proof:

$$\begin{aligned}
\Sigma_{A\vec{\mathbf{x}}} &= \mathrm{E}\left((A\vec{\mathbf{x}})\,(A\vec{\mathbf{x}})^T\right) - \mathrm{E}\,(A\vec{\mathbf{x}})\,\mathrm{E}\,(A\vec{\mathbf{x}})^T \\
&= A\left(\mathrm{E}\left(\vec{\mathbf{x}}\vec{\mathbf{x}}^T\right) - \mathrm{E}(\vec{\mathbf{x}})\mathrm{E}(\vec{\mathbf{x}})^T\right)A^T \\
&= A\Sigma_{\vec{\mathbf{x}}}A^T
\end{aligned}$$

Corollary: Variance in a fixed direction

The variance of $\vec{\mathbf{x}}$ in the direction of a unit-norm vector $\vec{v}$ equals

$$\mathrm{Var}\left(\vec{v}^{\,T}\vec{\mathbf{x}}\right) = \vec{v}^{\,T}\Sigma_{\vec{\mathbf{x}}}\vec{v}$$

Covariance matrix captures variance in every direction!

Also, covariance matrices are positive semidefinite, i.e. for any $\vec{v}$,

$$\vec{v}^{\,T}\Sigma_{\vec{\mathbf{x}}}\vec{v} \geq 0$$

SVD of covariance matrix

Because covariance matrices are symmetric and positive semidefinite

$$\Sigma_{\vec{\mathbf{x}}} = U \Lambda U^T$$

$$= \begin{bmatrix} \vec{u}_1 & \vec{u}_2 & \cdots & \vec{u}_n \end{bmatrix} \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & s_n \end{bmatrix} \begin{bmatrix} \vec{u}_1 & \vec{u}_2 & \cdots & \vec{u}_n \end{bmatrix}^T$$

Directions of maximum variance

The SVD of the covariance matrix $\Sigma_{\vec{\mathbf{x}}}$ of a random vector $\vec{\mathbf{x}}$ satisfies

$$
\begin{aligned}
s_1 &= \max_{||\vec{v}||_2=1} \operatorname{Var}\left(\vec{v}^{\,T}\vec{\mathbf{x}}\right) \\
\vec{u}_1 &= \arg\max_{||\vec{v}||_2=1} \operatorname{Var}\left(\vec{v}^{\,T}\vec{\mathbf{x}}\right) \\
s_k &= \max_{||\vec{v}||_2=1,\vec{v}\perp\vec{u}_1,\ldots,\vec{u}_{k-1}} \operatorname{Var}\left(\vec{v}^{\,T}\vec{\mathbf{x}}\right) \\
\vec{u}_k &= \arg\max_{||\vec{v}||_2=1,\vec{v}\perp\vec{u}_1,\ldots,\vec{u}_{k-1}} \operatorname{Var}\left(\vec{v}^{\,T}\vec{\mathbf{x}}\right)
\end{aligned}
$$

Proof

$$\begin{aligned}
\operatorname{Var}\left(\vec{v}^{\,T}\vec{\mathbf{x}}\right) &= \vec{v}^{\,T}\Sigma_{\vec{\mathbf{x}}}\vec{v} \\
&= \vec{v}^{\,T}USU^{T}\vec{v} \\
&= \left|\left|\sqrt{S}U^{T}\vec{v}\right|\right|_2^2
\end{aligned}$$

Singular values

The singular values of A satisfy

$$
\begin{aligned}
s_1 &= \max_{\left\{||\vec{x}||_2=1 \mid \vec{x}\in\mathbb{R}^n\right\}} ||A\vec{x}||_2 \\
&= \max_{\left\{||\vec{y}||_2=1 \mid \vec{y}\in\mathbb{R}^p\right\}} \left|\left|A^T\vec{y}\right|\right|_2 \\
s_i &= \max_{\left\{||\vec{x}||_2=1 \mid \vec{x}\in\mathbb{R}^n, \vec{x}\perp\vec{u}_1,\ldots,\vec{u}_{i-1}\right\}} ||A\vec{x}||_2 \\
&= \max_{\left\{||\vec{y}||_2=1 \mid \vec{y}\in\mathbb{R}^p, \vec{y}\perp\vec{v}_1,\ldots,\vec{v}_{i-1}\right\}} \left|\left|A^T\vec{y}\right|\right|_2, \qquad 2 \leq i \leq \min\left\{m, n\right\}
\end{aligned}
$$

Singular vectors

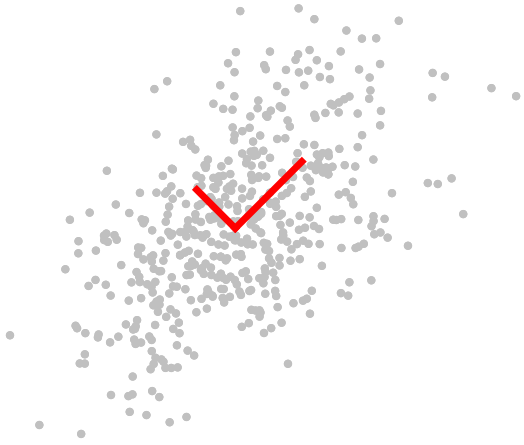
The left singular vectors of A satisfy

$$\vec{u}_1 = \underset{\left\{||\vec{y}||_2=1 \mid \vec{y}\in\mathbb{R}^p\right\}}{\arg\max} \left|\left|A^T\vec{y}\right|\right|_2$$

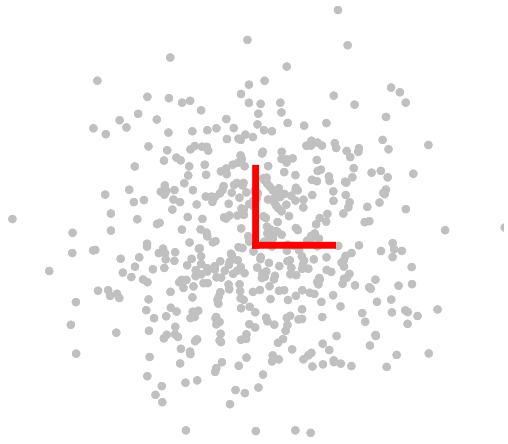
$$\vec{u}_i = \underset{\left\{||\vec{y}||_2=1 \mid \vec{y}\in\mathbb{R}^p, \vec{y}\perp\vec{v}_1,\ldots,\vec{v}_{i-1}\right\}}{\arg\max} \left|\left|A^T\vec{y}\right|\right|_2, \qquad 2 \leq i \leq n$$

Directions of maximum variance

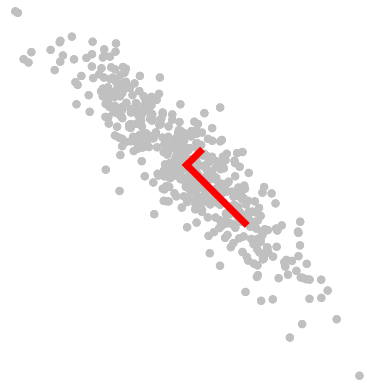
$\sqrt{s_1} = 1.22,\ \sqrt{s_2} = 0.71$



$\sqrt{s_1} = 1,\ \sqrt{s_2} = 1$



$\sqrt{s_1} = 1.38,\ \sqrt{s_2} = 0.32$



Sample covariance

For a data set (x_1, y_1), (x_2, y_2), $\ldots$, (x_n, y_n)

$$\mathrm{cov}\left((x_1, y_1), \ldots, (x_n, y_n)\right) := \frac{1}{n-1} \sum_{i=1}^{n} \left(x_i - \mathrm{av}\left(x_1, \ldots, x_n\right)\right)\left(y_i - \mathrm{av}\left(y_1, \ldots, y_n\right)\right)$$

If $(\mathbf{x_1}, \mathbf{y_1})$, $(\mathbf{x_2}, \mathbf{y_2})$, $\ldots$, $(\mathbf{x_n}, \mathbf{y_n})$ are iid samples from $\mathbf{x}$ and $\mathbf{y}$

$$\mathrm{E}\left(\mathrm{cov}\left((\mathbf{x_1}, \mathbf{y_1}), \ldots, (\mathbf{x_n}, \mathbf{y_n})\right)\right) = \mathrm{Cov}\left(\mathbf{x}, \mathbf{y}\right)$$

Sample covariance matrix

The sample covariance matrix of $\{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\} \in \mathbb{R}^p$

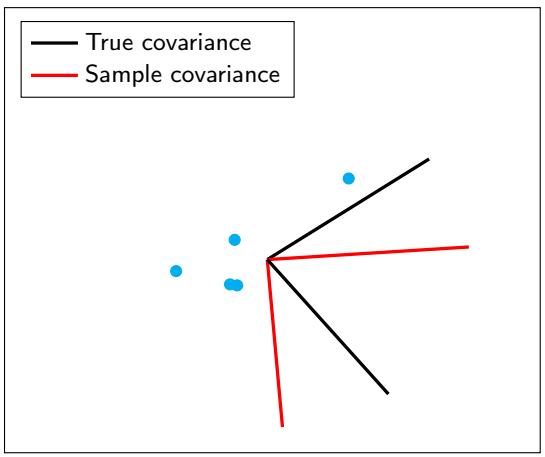
$$\Sigma\left(\vec{x}_1, \ldots, \vec{x}_n\right) := \frac{1}{n-1} \sum_{i=1}^{n} \left(\vec{x}_i - \text{av}\left(\vec{x}_1, \ldots, \vec{x}_n\right)\right) \left(\vec{x}_i - \text{av}\left(\vec{x}_1, \ldots, \vec{x}_n\right)\right)^T$$

$$\text{av}\left(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\right) := \frac{1}{n} \sum_{i=1}^{n} \vec{x}_i$$

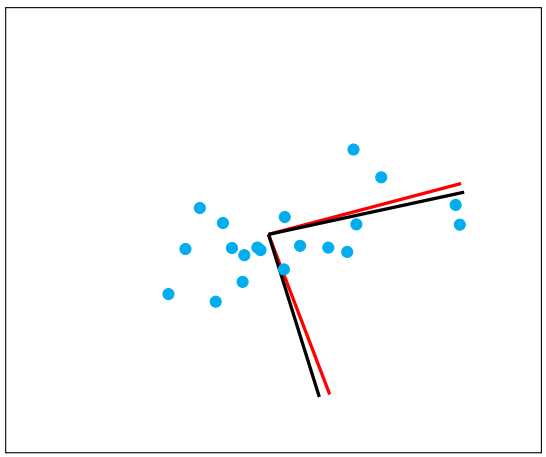
$$\Sigma\left(\vec{x}_1, \ldots, \vec{x}_n\right)_{ij} = \begin{cases} \text{var}\left(\vec{x}_1\left[i\right], \ldots, \vec{x}_n\left[i\right]\right) & \text{if } i = j, \\ \text{cov}\left(\left(\vec{x}_1\left[i\right], \vec{x}_1\left[j\right]\right), \ldots, \left(\vec{x}_n\left[i\right], \vec{x}_n\left[j\right]\right)\right) & \text{if } i \neq j \end{cases}$$

Sample covariance converges to true covariance

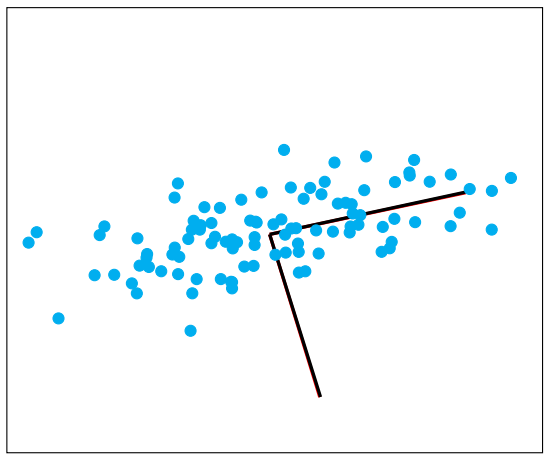
$n = 5$



$n = 20$



$n = 100$



Variation in a certain direction

For a unit vector $\vec{v} \in \mathbb{R}^p$

$$\text{var}\left(\vec{v}^{\,T}\vec{x}_1, \ldots, \vec{v}^{\,T}\vec{x}_n\right)$$

Variation in a certain direction

For a unit vector $\vec{v} \in \mathbb{R}^p$

$$\text{var}\left(\vec{v}^{\,T}\vec{x}_1, \ldots, \vec{v}^{\,T}\vec{x}_n\right)$$

$$= \frac{1}{n-1}\sum_{i=1}^{n}\left(\vec{v}^{\,T}\vec{x}_i - \text{av}\left(\vec{v}^{\,T}\vec{x}_1, \ldots, \vec{v}^{\,T}\vec{x}_n\right)\right)^2$$

Variation in a certain direction

For a unit vector $\vec{v} \in \mathbb{R}^p$

$$\begin{aligned}
&\text{var}\left(\vec{v}^{\,T}\vec{x}_1, \ldots, \vec{v}^{\,T}\vec{x}_n\right) \\
&= \frac{1}{n-1}\sum_{i=1}^{n}\left(\vec{v}^{\,T}\vec{x}_i - \text{av}\left(\vec{v}^{\,T}\vec{x}_1, \ldots, \vec{v}^{\,T}\vec{x}_n\right)\right)^2 \\
&= \frac{1}{n-1}\sum_{i=1}^{n}\left(\vec{v}^{\,T}\left(\vec{x}_i - \text{av}\left(\vec{x}_1, \ldots, \vec{x}_n\right)\right)\right)^2
\end{aligned}$$

Variation in a certain direction

For a unit vector $\vec{v} \in \mathbb{R}^p$

$$
\begin{aligned}
&\text{var}\left(\vec{v}^{\,T}\vec{x}_1, \ldots, \vec{v}^{\,T}\vec{x}_n\right) \\
&= \frac{1}{n-1}\sum_{i=1}^{n}\left(\vec{v}^{\,T}\vec{x}_i - \text{av}\left(\vec{v}^{\,T}\vec{x}_1, \ldots, \vec{v}^{\,T}\vec{x}_n\right)\right)^2 \\
&= \frac{1}{n-1}\sum_{i=1}^{n}\left(\vec{v}^{\,T}\left(\vec{x}_i - \text{av}\left(\vec{x}_1, \ldots, \vec{x}_n\right)\right)\right)^2 \\
&= \vec{v}^{\,T}\left(\frac{1}{n-1}\sum_{i=1}^{n}\left(\vec{x}_i - \text{av}\left(\vec{x}_1, \ldots, \vec{x}_n\right)\right)\left(\vec{x}_i - \text{av}\left(\vec{x}_1, \ldots, \vec{x}_n\right)\right)^T\right)\vec{v}
\end{aligned}
$$

Variation in a certain direction

For a unit vector $\vec{v} \in \mathbb{R}^p$

$$
\begin{aligned}
&\text{var}\left(\vec{v}^{\,T}\vec{x}_1, \ldots, \vec{v}^{\,T}\vec{x}_n\right) \\
&= \frac{1}{n-1}\sum_{i=1}^{n}\left(\vec{v}^{\,T}\vec{x}_i - \text{av}\left(\vec{v}^{\,T}\vec{x}_1, \ldots, \vec{v}^{\,T}\vec{x}_n\right)\right)^2 \\
&= \frac{1}{n-1}\sum_{i=1}^{n}\left(\vec{v}^{\,T}\left(\vec{x}_i - \text{av}\left(\vec{x}_1, \ldots, \vec{x}_n\right)\right)\right)^2 \\
&= \vec{v}^{\,T}\left(\frac{1}{n-1}\sum_{i=1}^{n}\left(\vec{x}_i - \text{av}\left(\vec{x}_1, \ldots, \vec{x}_n\right)\right)\left(\vec{x}_i - \text{av}\left(\vec{x}_1, \ldots, \vec{x}_n\right)\right)^T\right)\vec{v} \\
&= \vec{v}^{\,T}\Sigma\left(\vec{x}_1, \ldots, \vec{x}_n\right)\vec{v}
\end{aligned}
$$

Sample covariance matrix captures sample variance in every direction!

Principal component analysis

Given data vectors $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n \in \mathbb{R}^d$, compute the SVD of their sample covariance matrix

The left singular vectors are the principal directions

The principal values are the coefficients of the centered vectors in the basis of principal directions.

Equivalently

Center the data

$$\vec{c}_i = \vec{x}_i - \text{av}\left(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\right), \qquad 1 \leq i \leq n,$$

Compute the SVD of the matrix

$$C = \begin{bmatrix} \vec{c}_1 & \vec{c}_2 & \cdots & \vec{c}_n \end{bmatrix}$$

Result is the same because sample covariance matrix equals $\frac{1}{n-1} C C^T$

Directions of maximum variance

The principal directions satisfy

$$\vec{u}_1 = \underset{\left\{||\vec{v}||_2=1 \mid \vec{v}\in\mathbb{R}^n\right\}}{\arg\max} \operatorname{var}\left(\vec{v}^{\,T}\vec{x}_1, \ldots, \vec{v}^{\,T}\vec{x}_n\right)$$

$$\vec{u}_i = \underset{\left\{||\vec{v}||_2=1 \mid \vec{v}\in\mathbb{R}^n, \vec{v}\perp\vec{u}_1,\ldots,\vec{u}_{i-1}\right\}}{\arg\max} \operatorname{var}\left(\vec{v}^{\,T}\vec{x}_1, \ldots, \vec{v}^{\,T}\vec{x}_n\right), \qquad 2 \leq i \leq k$$

Directions of maximum variance

The associated singular values satisfy

$$\frac{s_1}{n-1} = \max_{\left\{||\vec{v}||_2=1 \mid \vec{v}\in\mathbb{R}^n\right\}} \text{var}\left(\vec{v}^{\,T}\vec{x}_1, \ldots, \vec{v}^{\,T}\vec{x}_n\right)$$

$$\frac{s_i}{n-1} = \max_{\left\{||\vec{v}||_2=1 \mid \vec{v}\in\mathbb{R}^n, \vec{v}\perp\vec{u}_1,\ldots,\vec{u}_{i-1}\right\}} \text{var}\left(\vec{v}^{\,T}\vec{x}_1, \ldots, \vec{v}^{\,T}\vec{x}_n\right), \quad 2 \leq i \leq k$$

Proof

For any vector $\vec{v}$

$$\text{var}\left(\vec{v}^{\,T}\vec{x}_1, \ldots, \vec{v}^{\,T}\vec{x}_n\right) = \vec{v}^{\,T}\Sigma\left(\vec{x}_1, \ldots, \vec{x}_n\right)\vec{v}$$

Singular values

The singular values of A satisfy

$$
\begin{aligned}
s_1 &= \max_{\left\{||\vec{x}||_2=1 \mid \vec{x}\in\mathbb{R}^n\right\}} ||A\vec{x}||_2 \\
&= \max_{\left\{||\vec{y}||_2=1 \mid \vec{y}\in\mathbb{R}^p\right\}} \left|\left|A^T\vec{y}\right|\right|_2 \\
s_i &= \max_{\left\{||\vec{x}||_2=1 \mid \vec{x}\in\mathbb{R}^n, \vec{x}\perp\vec{u}_1,\ldots,\vec{u}_{i-1}\right\}} ||A\vec{x}||_2 \\
&= \max_{\left\{||\vec{y}||_2=1 \mid \vec{y}\in\mathbb{R}^p, \vec{y}\perp\vec{v}_1,\ldots,\vec{v}_{i-1}\right\}} \left|\left|A^T\vec{y}\right|\right|_2, \qquad 2 \leq i \leq \min\left\{m, n\right\}
\end{aligned}
$$

Singular vectors

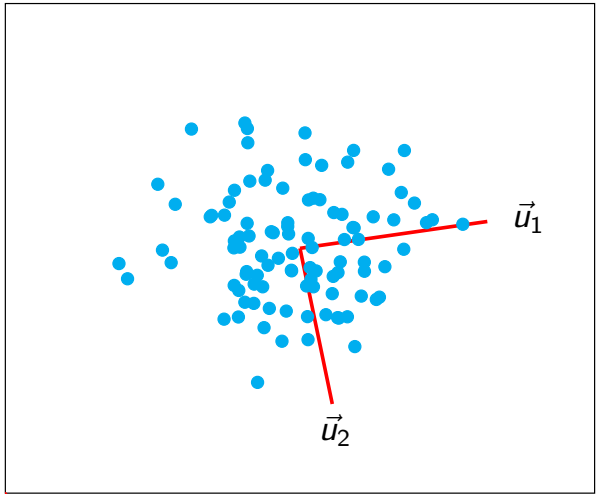
The left singular vectors of A satisfy

$$\vec{u}_1 = \underset{\left\{||\vec{y}||_2=1 \mid \vec{y}\in\mathbb{R}^p\right\}}{\arg\max} \left|\left|A^T\vec{y}\right|\right|_2$$

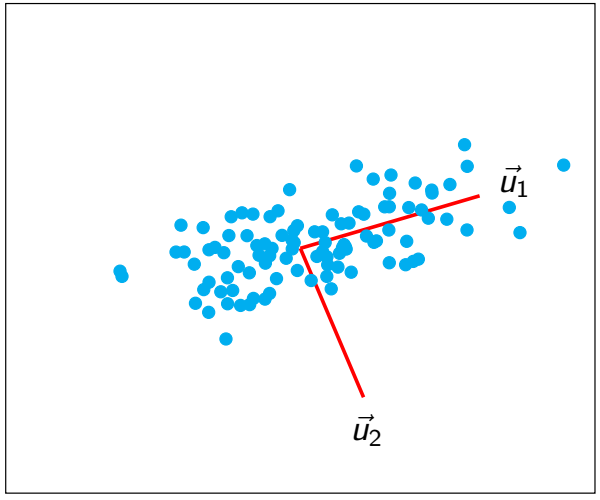
$$\vec{u}_i = \underset{\left\{||\vec{y}||_2=1 \mid \vec{y}\in\mathbb{R}^p, \vec{y}\perp\vec{v}_1,\ldots,\vec{v}_{i-1}\right\}}{\arg\max} \left|\left|A^T\vec{y}\right|\right|_2, \qquad 2 \leq i \leq n$$

PCA in 2D

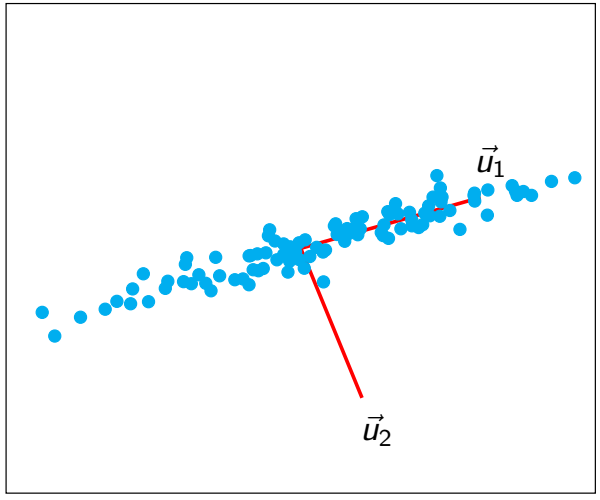
$\sqrt{s_1/(n-1)} = 0.705,$
$\sqrt{s_2/(n-1)} = 0.690$



$\sqrt{s_1/(n-1)} = 0.983,$
$\sqrt{s_2/(n-1)} = 0.356$



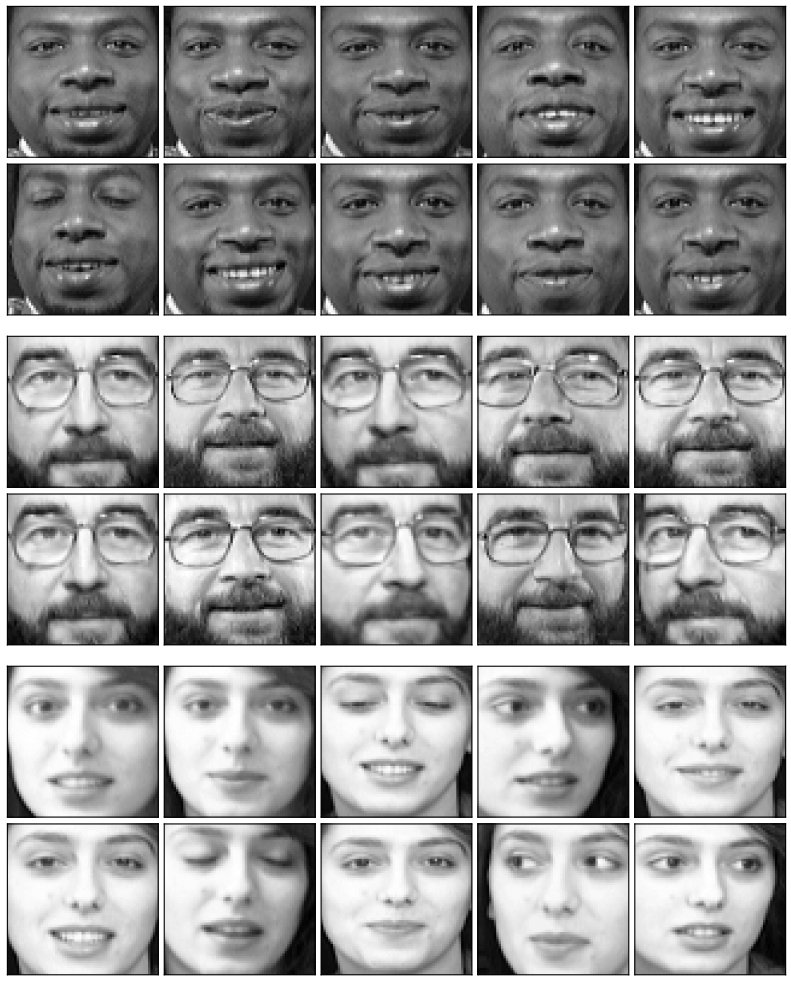
$\sqrt{s_1/(n-1)} = 1.349,$
$\sqrt{s_2/(n-1)} = 0.144$



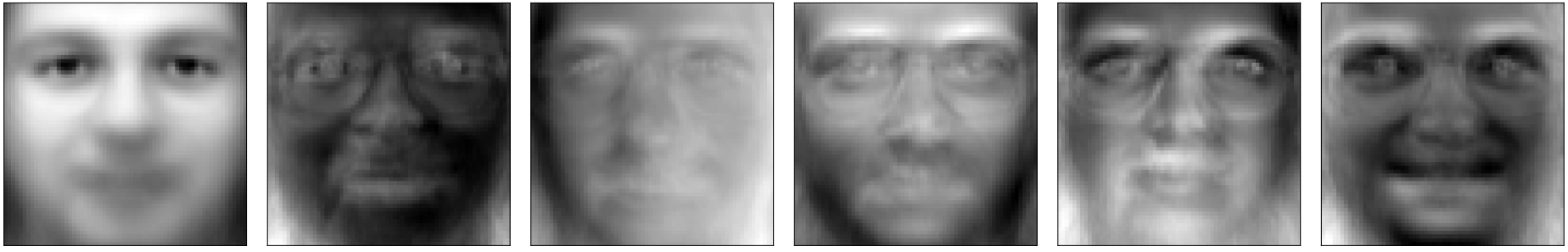
PCA of faces

Data set of 400 64 $\times$ 64 images from 40 subjects (10 per subject)

Each face is vectorized and interpreted as a vector in $\mathbb{R}^{4096}$

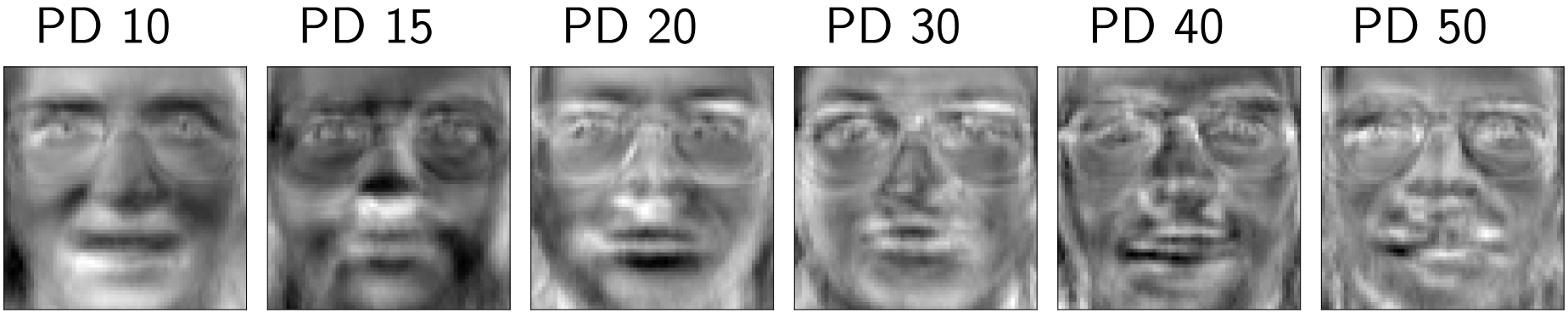


PCA of faces

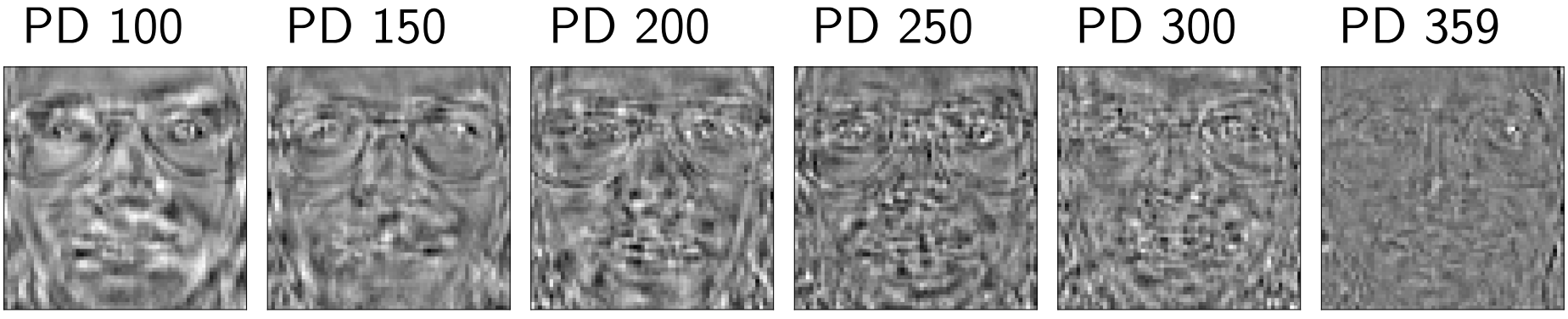


$\sqrt{s_i/(n-1)}$	330	251	192	152	130

PCA of faces



PCA of faces



Projection onto first 7 principal directions

$=$ 8613

Center

- 2459

PD 1

$+$ 665

PD 2

- 180

PD 3

$+$ 301

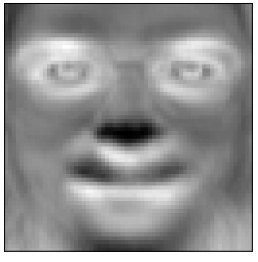
PD 4

$+$ 566

PD 5

$+$ 638

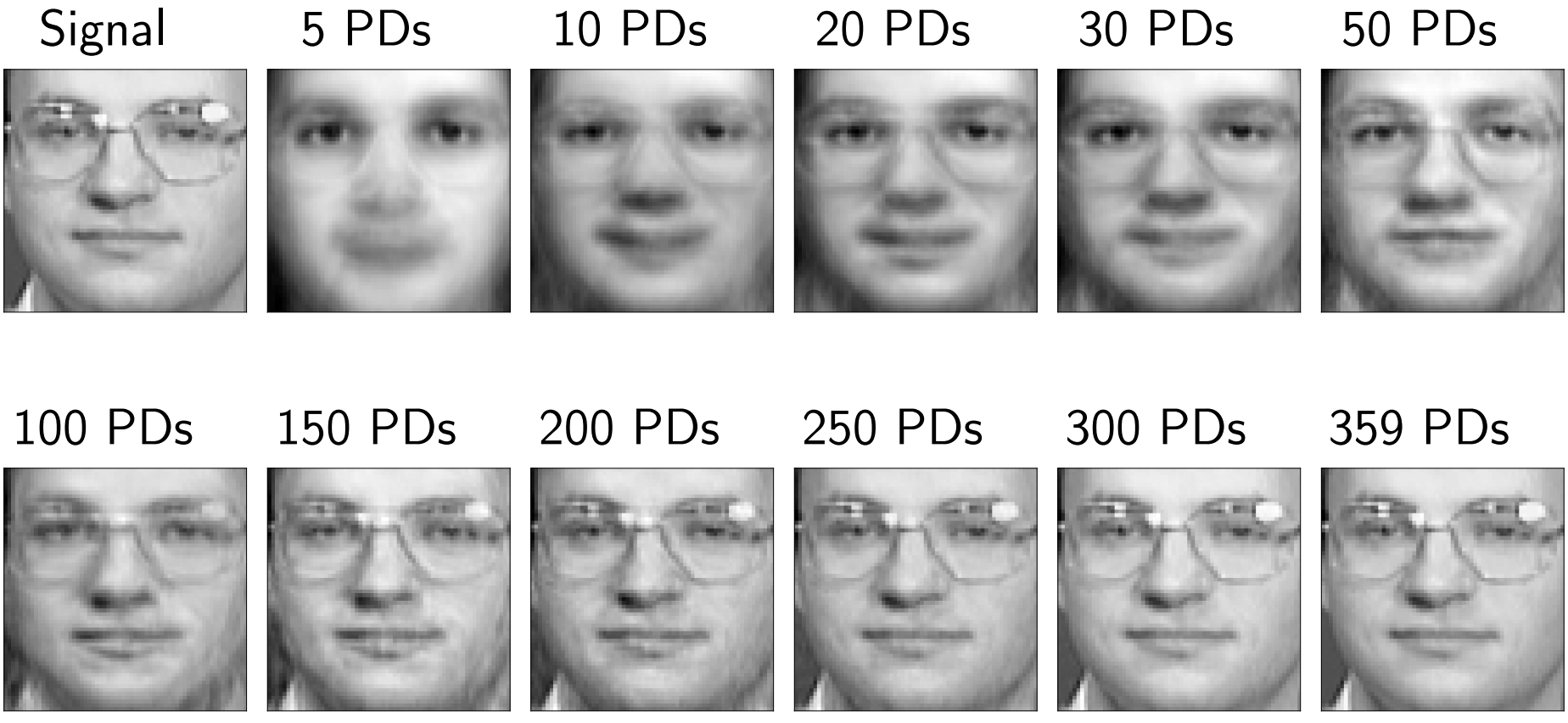
PD 6



$+$ 403

PD 7

Projection onto first k principal directions



Motivating applications

The singular-value decomposition

Principal component analysis

Dimensionality reduction

Linear regression

Ridge regression

Low-rank matrix estimation

Dimensionality reduction

Data with a large number of features can be difficult to analyze or process

Dimensionality reduction is a useful preprocessing step

If data are modeled as vectors in $\mathbb{R}^p$ we can reduce the dimension by projecting onto $\mathbb{R}^k$, where $k < p$

For orthogonal projections, the new representation is $\langle \vec{b}_1, \vec{x} \rangle$, $\langle \vec{b}_2, \vec{x} \rangle$, ..., $\langle \vec{b}_k, \vec{x} \rangle$ for a basis $\vec{b}_1$, ..., $\vec{b}_k$ of the subspace that we project on

Problem: How do we choose the subspace?

Optimal subspace for orthogonal projection

Given a set of vectors $\vec{a}_1, \vec{a}_2, \ldots \vec{a}_n$ and a fixed dimension $k \leq n$, the SVD of

$$A := \begin{bmatrix} \vec{a}_1 & \vec{a}_2 & \cdots & \vec{a}_n \end{bmatrix} \in \mathbb{R}^{m \times n}$$

provides the k-dimensional subspace that captures the most energy

$$\sum_{i=1}^{n} \left|\left| \mathcal{P}_{\text{span}(\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k)} \, \vec{a}_i \right|\right|_2^2 \geq \sum_{i=1}^{n} ||\mathcal{P}_{\mathcal{S}} \, \vec{a}_i||_2^2$$

for any subspace $\mathcal{S}$ of dimension k

Proof

Because $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k$ are orthonormal

$$\sum_{i=1}^{n} \left|\left|\mathcal{P}_{\text{span}(\vec{u}_1,\vec{u}_2,\ldots,\vec{u}_k)}\, \vec{a}_i\right|\right|_2^2 = \sum_{i=1}^{n}\sum_{j=1}^{k} \langle \vec{u}_j, \vec{a}_i \rangle^2$$

Proof

Because $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k$ are orthonormal

$$\sum_{i=1}^{n} \left|\left|\mathcal{P}_{\text{span}(\vec{u}_1,\vec{u}_2,\ldots,\vec{u}_k)}\, \vec{a}_i\right|\right|_2^2 = \sum_{i=1}^{n}\sum_{j=1}^{k} \langle \vec{u}_j, \vec{a}_i \rangle^2$$

$$= \sum_{j=1}^{k} \left|\left| A^T \vec{u}_j \right|\right|_2^2$$

Proof

Because $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k$ are orthonormal

$$\sum_{i=1}^{n} \left|\left| \mathcal{P}_{\text{span}(\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k)} \, \vec{a}_i \right|\right|_2^2 = \sum_{i=1}^{n} \sum_{j=1}^{k} \langle \vec{u}_j, \vec{a}_i \rangle^2$$

$$= \sum_{j=1}^{k} \left|\left| A^T \vec{u}_j \right|\right|_2^2$$

Induction on k

Proof

Because $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k$ are orthonormal

$$\sum_{i=1}^{n} \left|\left| \mathcal{P}_{\text{span}(\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k)} \, \vec{a}_i \right|\right|_2^2 = \sum_{i=1}^{n} \sum_{j=1}^{k} \langle \vec{u}_j, \vec{a}_i \rangle^2$$

$$= \sum_{j=1}^{k} \left|\left| A^T \vec{u}_j \right|\right|_2^2$$

Induction on k

The base case $k = 1$ follows from

$$\vec{u}_1 = \underset{\left\{ ||\vec{y}||_2 = 1 \mid \vec{y} \in \mathbb{R}^m \right\}}{\arg\max} \left|\left| A^T \vec{y} \right|\right|_2$$

Proof

Let $\mathcal{S}$ be a subspace of dimension k

$\mathcal{S} \cap \text{span}\left(\vec{u}_1, \ldots, \vec{u}_{k-1}\right)^{\perp}$ contains a nonzero vector $\vec{b}$

If $\dim\left(\mathcal{V}\right)$ *has dimension* n, $\mathcal{S}_1, \mathcal{S}_2 \subseteq \mathcal{V}$ *and* $\dim\left(\mathcal{S}_1\right) + \dim\left(\mathcal{S}_2\right) > n$, *then* $\dim\left(\mathcal{S}_1 \cap \mathcal{S}_2\right) \geq 1$

Proof

Let $\mathcal{S}$ be a subspace of dimension k

$\mathcal{S} \cap \text{span}\left(\vec{u}_1, \ldots, \vec{u}_{k-1}\right)^{\perp}$ contains a nonzero vector $\vec{b}$

If $\dim\left(\mathcal{V}\right)$ *has dimension* n, $\mathcal{S}_1, \mathcal{S}_2 \subseteq \mathcal{V}$ *and* $\dim\left(\mathcal{S}_1\right) + \dim\left(\mathcal{S}_2\right) > n$, *then* $\dim\left(\mathcal{S}_1 \cap \mathcal{S}_2\right) \geq 1$

There exists an orthonormal basis $\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_k$ for $\mathcal{S}$ such that $\vec{b}_k := \vec{b}$ is orthogonal to $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_{k-1}$

Induction hypothesis

$$\sum_{i=1}^{k-1} \left|\left| A^T \vec{u}_i \right|\right|_2^2 = \sum_{i=1}^{n} \left|\left| \mathcal{P}_{\text{span}(\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_{k-1})} \vec{a}_i \right|\right|_2^2$$

$$\geq \sum_{i=1}^{n} \left|\left| \mathcal{P}_{\text{span}\left(\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_{k-1}\right)} \vec{a}_i \right|\right|_2^2$$

$$= \sum_{i=1}^{k-1} \left|\left| A^T \vec{b}_i \right|\right|_2^2$$

Proof

Recall that

$$\vec{u}_k = \underset{\left\{||\vec{y}||_2=1 \mid \vec{y}\in\mathbb{R}^m, \vec{y}\perp\vec{u}_1,\ldots,\vec{u}_{k-1}\right\}}{\arg\max} \left|\left|A^T\vec{y}\right|\right|_2$$

which implies

$$\left|\left|A^T\vec{u}_k\right|\right|_2^2 \geq \left|\left|A^T\vec{b}_k\right|\right|_2^2$$

Proof

Recall that

$$\vec{u}_k = \underset{\left\{||\vec{y}||_2=1 \mid \vec{y}\in\mathbb{R}^m, \vec{y}\perp\vec{u}_1,\ldots,\vec{u}_{k-1}\right\}}{\arg\max} \left|\left|A^T\vec{y}\right|\right|_2$$

which implies

$$\left|\left|A^T\vec{u}_k\right|\right|_2^2 \geq \left|\left|A^T\vec{b}_k\right|\right|_2^2$$

We conclude

$$\begin{aligned}\sum_{i=1}^{n} \left|\left|\mathcal{P}_{\text{span}(\vec{u}_1,\vec{u}_2,\ldots,\vec{u}_k)}\,\vec{a}_i\right|\right|_2^2 &= \sum_{i=1}^{k} \left|\left|A^T\vec{u}_i\right|\right|_2^2 \\ &\geq \sum_{i=1}^{k} \left|\left|A^T\vec{b}_i\right|\right|_2^2 \\ &= \sum_{i=1}^{n} ||\mathcal{P}_S\,\vec{a}_i||_2^2\end{aligned}$$

Nearest-neighbor classification

Training set of points and labels $\{\vec{x}_1, l_1\}, \ldots, \{\vec{x}_n, l_n\}$

To classify a new data point $\vec{y}$, find

$$i^* := \arg \min_{1 \leq i \leq n} ||\vec{y} - \vec{x}_i||_2 ,$$

and assign l_{i^*} to $\vec{y}$

Cost: $\mathcal{O}(mnp)$ to classify m new points

Nearest neighbors in principal-component space

Idea: Project onto first k main principal directions beforehand

Cost:

- $\mathcal{O}\left(p^2 n\right)$, if $p < n$, to compute principal dimensions
- knp operations to project training set
- kmp operations to project test set
- kmn to perform nearest-neighbor classification

Faster if $m > p$

Face recognition

Training set: 360 64×64 images from 40 different subjects (9 each)

Test set: 1 new image from each subject

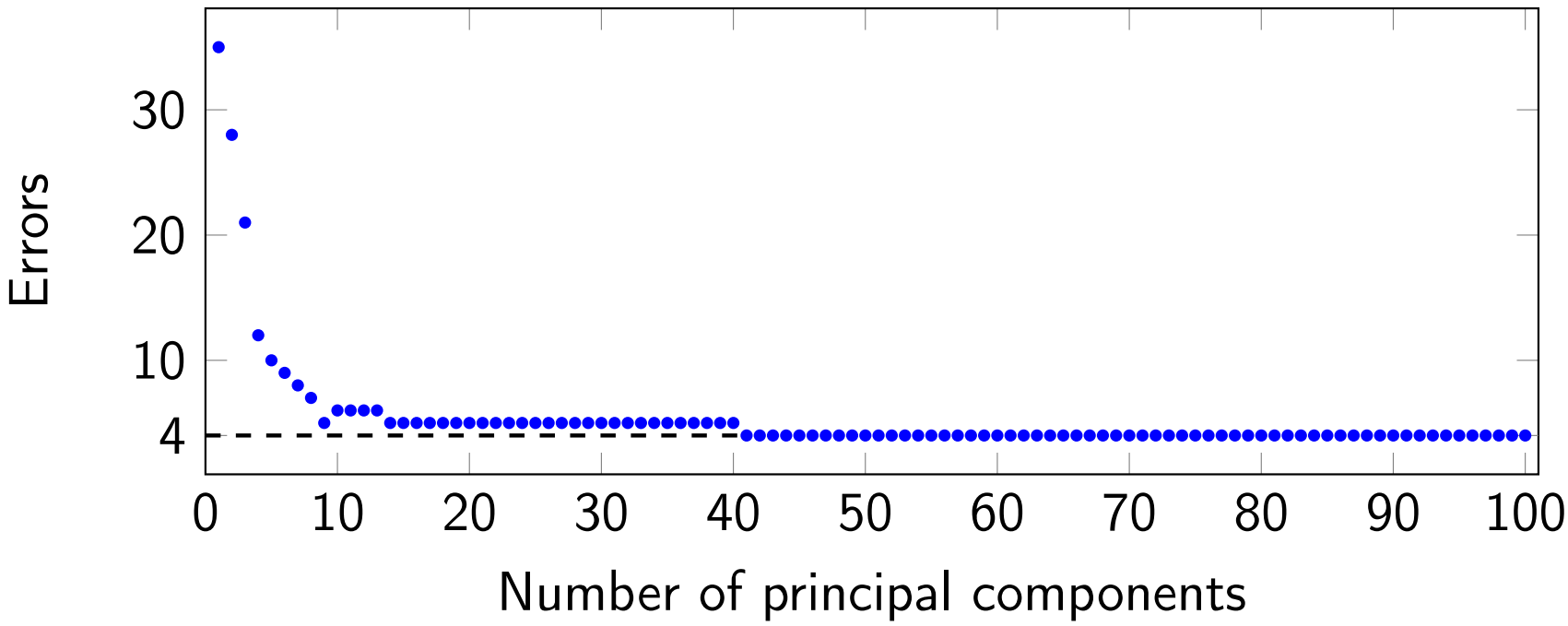
We model each image as a vector in $\mathbb{R}^{4096}$ ($m = 4096$)

To classify we:

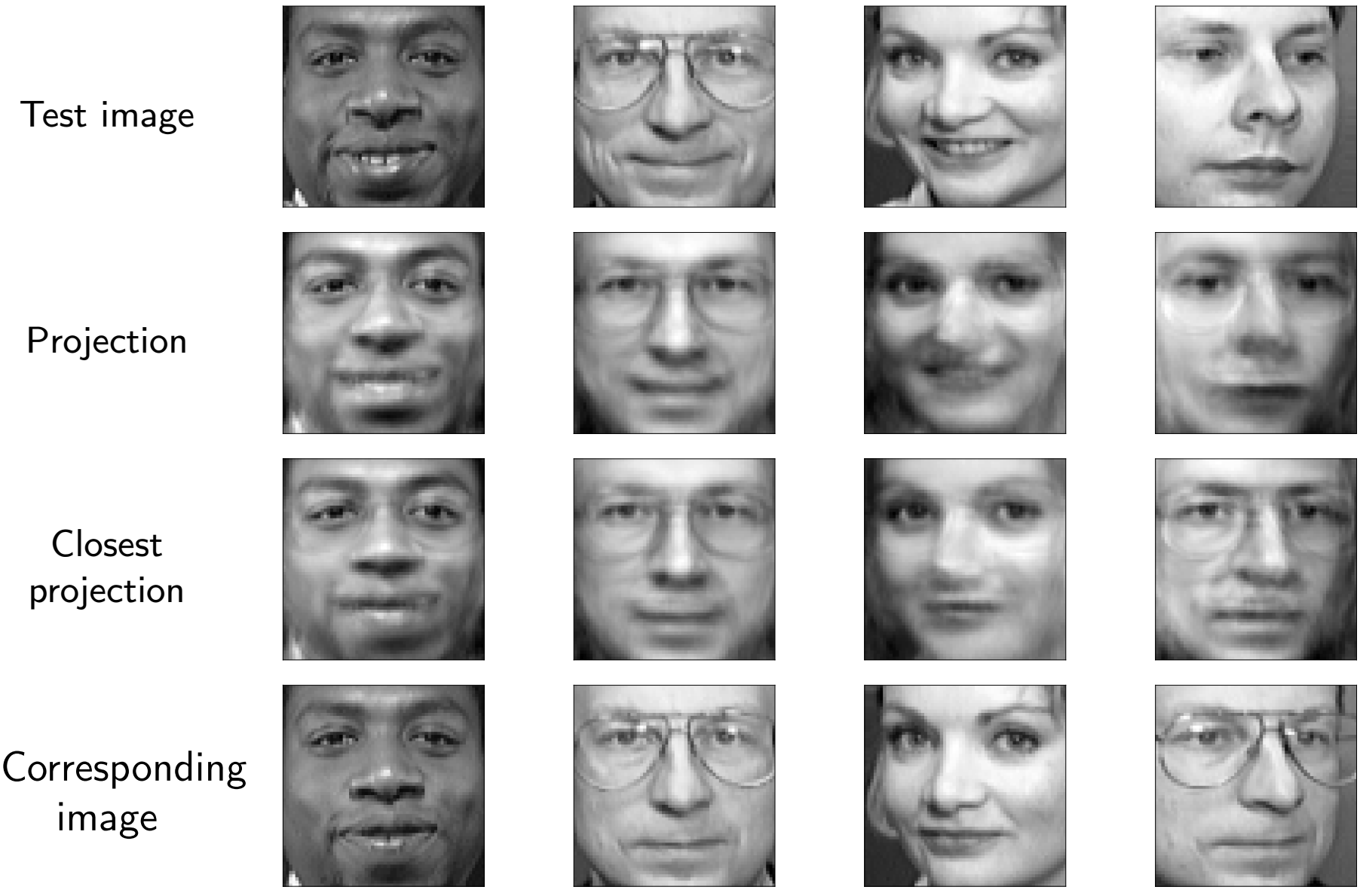
1. Project onto first k principal directions

2. Apply nearest-neighbor classification using the ℓ_2-norm distance in $\mathbb{R}^k$

Performance



Nearest neighbor in $\mathbb{R}^{41}$



Dimensionality reduction for visualization

Motivation: Visualize high-dimensional features projected onto 2D or 3D

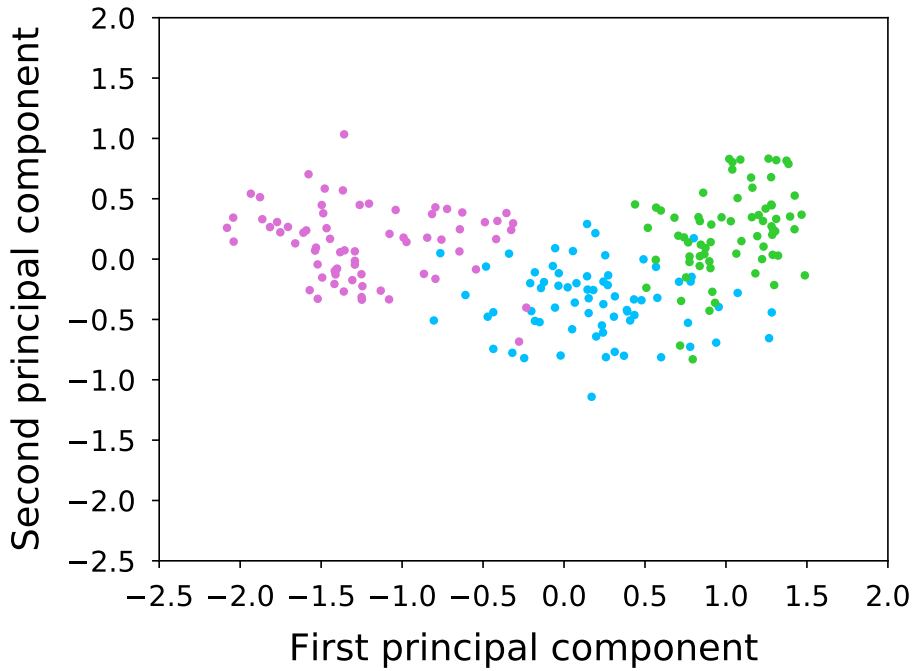
Example:

Seeds from three different varieties of wheat: Kama, Rosa and Canadian

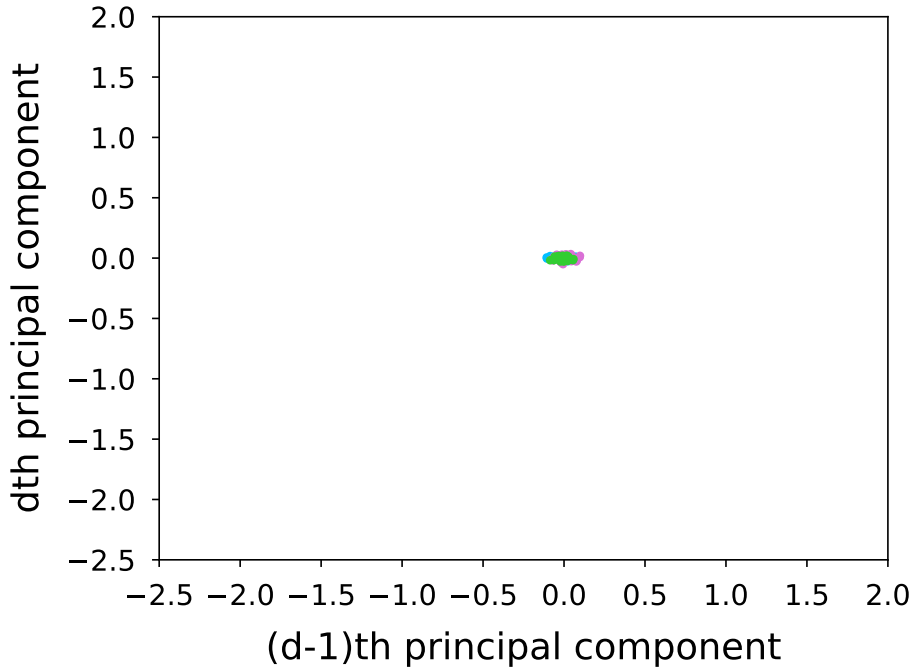
Features:

- Area
- Perimeter
- Compactness
- Length of kernel
- Width of kernel
- Asymmetry coefficient
- Length of kernel groove

Projection onto two first PDs



Projection onto two last PDs



Motivating applications

The singular-value decomposition

Principal component analysis

Dimensionality reduction

Linear regression

Ridge regression

Low-rank matrix estimation

Regression

The aim is to learn a function h that relates

- a response or dependent variable y
- to several observed variables $\vec{x} \in \mathbb{R}^p$, known as covariates, features or independent variables

The response is assumed to be of the form

$$y \approx h(\vec{x})$$

Linear regression

The regression function h is assumed to be linear

$$y^{(i)} \approx \vec{x}^{(i)\,T} \vec{\beta} + \beta_0, \quad 1 \leq i \leq n$$

We estimate $\vec{\beta} \in \mathbb{R}^p$ from training dataset

$$\mathcal{S}_{\text{train}} := \left\{ \left(y^{(1)}, \vec{x}^{(1)} \right), \left(y^{(2)}, \vec{x}^{(2)} \right), \ldots, \left(y^{(n)}, \vec{x}^{(n)} \right) \right\}$$

Linear regression

In matrix form

$$\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \dots \\ y^{(n)} \end{bmatrix} \approx \begin{bmatrix} \vec{x}^{(1)}[1] & \vec{x}^{(1)}[2] & \dots & \vec{x}^{(1)}[p] \\ \vec{x}^{(2)}[1] & \vec{x}^{(2)}[2] & \dots & \vec{x}^{(2)}[p] \\ \dots & \dots & \dots & \dots \\ \vec{x}^{(n)}[1] & \vec{x}^{(n)}[2] & \dots & \vec{x}^{(n)}[p] \end{bmatrix} \begin{bmatrix} \vec{\beta}[1] \\ \vec{\beta}[2] \\ \dots \\ \vec{\beta}[p] \end{bmatrix}.$$

Equivalently,

$$\vec{y} \approx X\vec{\beta}$$

Linear model for GDP

State	GDP (millions)	Population	Unemployment Rate
North Dakota	52 089	757 952	2.4
Alabama	204 861	4 863 300	3.8
Mississippi	107 680	2 988 726	5.2
Arkansas	120 689	2 988 248	3.5
Kansas	153 258	2 907 289	3.8
Georgia	525 360	10 310 371	4.5
Iowa	178 766	3 134 693	3.2
West Virginia	73 374	1 831 102	5.1
Kentucky	197 043	4 436 974	5.2
Tennessee	???	6 651 194	3.0

Centering

$$\vec{y}_{\text{cent}} = \begin{bmatrix} -127\ 147 \\ 25\ 625 \\ -71\ 556 \\ -58\ 547 \\ -25\ 978 \\ 470 \\ -105\ 862 \\ 17\ 807 \end{bmatrix} \qquad X_{\text{cent}} = \begin{bmatrix} 3\ 044\ 121 & -1.7 \\ 1\ 061\ 227 & -2.8 \\ -813\ 346 & 1.1 \\ -813\ 825 & -5.8 \\ -894\ 784 & -2.8 \\ 6508\ 298 & 4.2 \\ -667\ 379 & -8.8 \\ -1\ 970\ 971 & 1.0 \\ 634\ 901 & 1.1 \end{bmatrix}$$

$$\text{av}\,(\vec{y}) = 179\ 236 \qquad \text{av}\,(X) = \begin{bmatrix} 3\ 802\ 073 & 4.1 \end{bmatrix}$$

Normalizing

$$\vec{y}_{\text{norm}} = \begin{bmatrix} -0.321 \\ 0.065 \\ -0.180 \\ -0.148 \\ -0.065 \\ 0.872 \\ -0.001 \\ -0.267 \\ 0.045 \end{bmatrix} \qquad X_{\text{norm}} = \begin{bmatrix} -0.394 & -0.600 \\ 0.137 & -0.099 \\ -0.105 & 0.401 \\ -0.105 & -0.207 \\ -0.116 & -0.099 \\ 0.843 & 0.151 \\ -0.086 & -0.314 \\ -0.255 & 0.366 \\ 0.082 & 0.401 \end{bmatrix}$$

$$\text{std}\,(\vec{y}) = 396\ 701 \qquad \text{std}\,(X) = \begin{bmatrix} 7\ 720\ 656 & 2.80 \end{bmatrix}$$

Linear model for GDP

Aim: find $\vec{\beta} \in \mathbb{R}^2$ such that $\vec{y}_{\text{norm}} \approx X_{\text{norm}}\, \vec{\beta}$

The estimate for the GDP of Tennessee will be

$$\vec{y}^{\,\text{Ten}} = \text{av}\,(\vec{y}) + \text{std}\,(\vec{y}) \left\langle \vec{x}^{\,\text{Ten}}_{\text{norm}}, \vec{\beta} \right\rangle$$

where $\vec{x}^{\,\text{Ten}}_{\text{norm}}$ is centered using $\text{av}\,(X)$ and normalized using $\text{std}\,(X)$

Least squares

For fixed $\vec{\beta}$ we can evaluate the error using

$$\sum_{i=1}^{n} \left(y^{(i)} - \vec{x}^{(i)\,T} \vec{\beta} \right)^2 = \left|\left| \vec{y} - X\vec{\beta} \right|\right|_2^2$$

The least-squares estimate $\vec{\beta}_{\mathrm{LS}}$ minimizes this cost function

$$\vec{\beta}_{\mathrm{LS}} := \arg\min_{\vec{\beta}} \quad \left|\left| \vec{y} - X\vec{\beta} \right|\right|_2$$

$$= \left(X^T X \right)^{-1} X^T \vec{y}$$

if X is full rank and $n \geq p$

Least-squares solution

Let $X = USV^T$

$$\vec{y} = UU^T\vec{y} + \left(I - UU^T\right)\vec{y}$$

By the Pythagorean theorem

$$\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 =$$

Least-squares solution

Let $X = USV^T$

$$\vec{y} = UU^T\vec{y} + \left(I - UU^T\right)\vec{y}$$

By the Pythagorean theorem

$$\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 = \left|\left|\left(I - UU^T\right)\vec{y}\right|\right|_2^2 + \left|\left|UU^T\vec{y} - X\vec{\beta}\right|\right|_2^2$$

$$\arg\min_{\vec{\beta}} \quad \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2$$

Least-squares solution

Let $X = USV^T$

$$\vec{y} = UU^T\vec{y} + \left(I - UU^T\right)\vec{y}$$

By the Pythagorean theorem

$$\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 = \left|\left|\left(I - UU^T\right)\vec{y}\right|\right|_2^2 + \left|\left|UU^T\vec{y} - X\vec{\beta}\right|\right|_2^2$$

$$\arg\min_{\vec{\beta}} \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 = \arg\min_{\vec{\beta}} \left|\left|UU^T\vec{y} - X\vec{\beta}\right|\right|_2^2$$

Least-squares solution

Let $X = USV^T$

$$\vec{y} = UU^T\vec{y} + \left(I - UU^T\right)\vec{y}$$

By the Pythagorean theorem

$$\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 = \left|\left|\left(I - UU^T\right)\vec{y}\right|\right|_2^2 + \left|\left|UU^T\vec{y} - X\vec{\beta}\right|\right|_2^2$$

$$\begin{aligned}\arg\min_{\vec{\beta}} \quad \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 &= \arg\min_{\vec{\beta}} \quad \left|\left|UU^T\vec{y} - X\vec{\beta}\right|\right|_2^2 \\ &= \arg\min_{\vec{\beta}} \quad \left|\left|UU^T\vec{y} - USV^T\vec{\beta}\right|\right|_2^2\end{aligned}$$

Least-squares solution

Let $X = USV^T$

$$\vec{y} = UU^T\vec{y} + \left(I - UU^T\right)\vec{y}$$

By the Pythagorean theorem

$$\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 = \left|\left|\left(I - UU^T\right)\vec{y}\right|\right|_2^2 + \left|\left|UU^T\vec{y} - X\vec{\beta}\right|\right|_2^2$$

$$\begin{aligned}
\arg\min_{\vec{\beta}} \quad \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 &= \arg\min_{\vec{\beta}} \quad \left|\left|UU^T\vec{y} - X\vec{\beta}\right|\right|_2^2 \\
&= \arg\min_{\vec{\beta}} \quad \left|\left|UU^T\vec{y} - USV^T\vec{\beta}\right|\right|_2^2 \\
&= \arg\min_{\vec{\beta}} \quad \left|\left|U^T\vec{y} - SV^T\vec{\beta}\right|\right|_2^2
\end{aligned}$$

Least-squares solution

Let $X = USV^T$

$$\vec{y} = UU^T\vec{y} + \left(I - UU^T\right)\vec{y}$$

By the Pythagorean theorem

$$\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 = \left|\left|\left(I - UU^T\right)\vec{y}\right|\right|_2^2 + \left|\left|UU^T\vec{y} - X\vec{\beta}\right|\right|_2^2$$

$$\begin{aligned}
\arg\min_{\vec{\beta}} \quad \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 &= \arg\min_{\vec{\beta}} \quad \left|\left|UU^T\vec{y} - X\vec{\beta}\right|\right|_2^2 \\
&= \arg\min_{\vec{\beta}} \quad \left|\left|UU^T\vec{y} - USV^T\vec{\beta}\right|\right|_2^2 \\
&= \arg\min_{\vec{\beta}} \quad \left|\left|U^T\vec{y} - SV^T\vec{\beta}\right|\right|_2^2 \\
&= VS^{-1}U^T\vec{y} = \left(X^TX\right)^{-1}X^T\vec{y}
\end{aligned}$$

Linear model for GDP

The least-squares estimate is

$$\vec{\beta}_{\mathsf{LS}} = \begin{bmatrix} 1.019 \\ -0.111 \end{bmatrix}$$

GDP seems to be proportional to population and inversely proportional to unemployment

Linear model for GDP

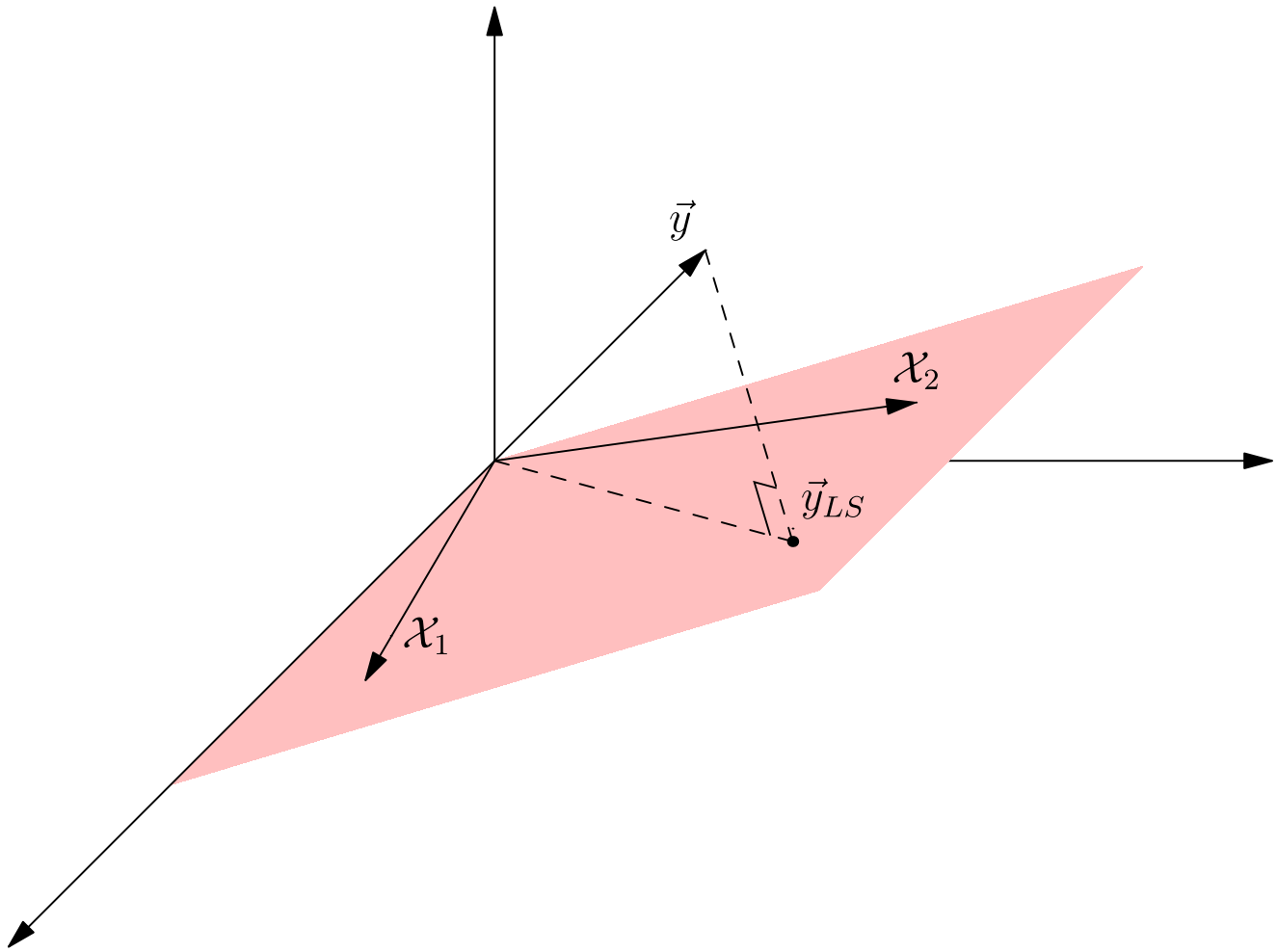
State	GDP	Estimate
North Dakota	52 089	46 241
Alabama	204 861	239 165
Mississippi	107 680	119 005
Arkansas	120 689	145 712
Kansas	153 258	136 756
Georgia	525 360	513 343
Iowa	178 766	158 097
West Virginia	73 374	59 969
Kentucky	197 043	194 829
Tennessee	328 770	345 352

Geometric interpretation

- Any vector $X\vec{\beta}$ is in the span of the columns of X
- The least-squares estimate is the closest vector to $\vec{y}$ that can be represented in this way
- This is the projection of $\vec{y}$ onto the column space of X

$$\begin{aligned} X\vec{\beta}_{\mathsf{LS}} &= XVS^{-1}U^T\vec{y} \\ &= USV^TVS^{-1}U^T\vec{y} \\ &= UU^T\vec{y} \end{aligned}$$

Geometric interpretation



Probabilistic interpretation

Let $\mathbf{y}$ be a random scalar with zero mean and $\vec{\mathbf{x}}$ a p-dimensional random vector

Goal: Estimate $\mathbf{y}$ as linear function of $\vec{\mathbf{x}}$

Criterion: Mean square error

$$\mathrm{MSE}(\vec{\beta}) := \mathrm{E}((\mathbf{y} - \vec{\mathbf{x}}^T \vec{\beta})^2)$$

Probabilistic interpretation

Let $\mathbf{y}$ be a random scalar with zero mean and $\vec{\mathbf{x}}$ a p-dimensional random vector

Goal: Estimate $\mathbf{y}$ as linear function of $\vec{\mathbf{x}}$

Criterion: Mean square error

$$\begin{aligned}\mathrm{MSE}(\vec{\beta}) &:= \mathrm{E}((\mathbf{y} - \vec{\mathbf{x}}^T \vec{\beta})^2) \\ &= \mathrm{E}\left(\mathbf{y}^2\right) - 2\mathrm{E}\left(\mathbf{y}\vec{\mathbf{x}}\right)^T \vec{\beta} + \vec{\beta}^T \mathrm{E}\left(\vec{\mathbf{x}}\vec{\mathbf{x}}^T\right) \vec{\beta}\end{aligned}$$

Probabilistic interpretation

Let $\mathbf{y}$ be a random scalar with zero mean and $\vec{\mathbf{x}}$ a p-dimensional random vector

Goal: Estimate $\mathbf{y}$ as linear function of $\vec{\mathbf{x}}$

Criterion: Mean square error

$$\begin{aligned}\mathrm{MSE}(\vec{\beta}) &:= \mathrm{E}((\mathbf{y} - \vec{\mathbf{x}}^T \vec{\beta})^2) \\ &= \mathrm{E}\left(\mathbf{y}^2\right) - 2\mathrm{E}\left(\mathbf{y}\vec{\mathbf{x}}\right)^T \vec{\beta} + \vec{\beta}^T \mathrm{E}\left(\vec{\mathbf{x}}\vec{\mathbf{x}}^T\right) \vec{\beta} \\ &= \mathrm{Var}\left(\mathbf{y}\right) - 2\Sigma_{\mathbf{y}\vec{\mathbf{x}}}^T \vec{\beta} + \vec{\beta}^T \Sigma_{\vec{\mathbf{x}}} \vec{\beta}\end{aligned}$$

Probabilistic interpretation

$$\nabla \mathrm{MSE}(\vec{\beta}) = 2\Sigma_{\vec{\mathbf{x}}}\vec{\beta} - 2\Sigma_{\mathbf{y}\vec{\mathbf{x}}}$$

Probabilistic interpretation

$$\nabla \mathrm{MSE}(\vec{\beta}) = 2\Sigma_{\vec{\mathbf{x}}}\vec{\beta} - 2\Sigma_{\mathbf{y}\vec{\mathbf{x}}}$$

$$\vec{\beta}_{\mathrm{MMSE}} = \Sigma_{\vec{\mathbf{x}}}^{-1}\Sigma_{\mathbf{y}\vec{\mathbf{x}}}$$

Probabilistic interpretation

$$\nabla \text{MSE}(\vec{\beta}) = 2\Sigma_{\vec{\mathbf{x}}}\vec{\beta} - 2\Sigma_{\mathbf{y}\vec{\mathbf{x}}}$$

$$\vec{\beta}_{\text{MMSE}} = \Sigma_{\vec{\mathbf{x}}}^{-1}\Sigma_{\mathbf{y}\vec{\mathbf{x}}}$$

$$\Sigma_{\vec{\mathbf{x}}} \approx \frac{1}{n}X^T X$$

$$\Sigma_{\mathbf{y}\vec{\mathbf{x}}} \approx \frac{1}{n}X\vec{y}$$

Probabilistic interpretation

$$\nabla \mathrm{MSE}(\vec{\beta}) = 2\Sigma_{\vec{\mathbf{x}}}\vec{\beta} - 2\Sigma_{\mathbf{y}\vec{\mathbf{x}}}$$

$$\vec{\beta}_{\mathrm{MMSE}} = \Sigma_{\vec{\mathbf{x}}}^{-1}\Sigma_{\mathbf{y}\vec{\mathbf{x}}}$$

$$\Sigma_{\vec{\mathbf{x}}} \approx \frac{1}{n}X^T X$$

$$\Sigma_{\mathbf{y}\vec{\mathbf{x}}} \approx \frac{1}{n}X\vec{y}$$

$$\vec{\beta}_{\mathrm{MMSE}} \approx \left(X^T X\right)^{-1} X\vec{y}$$

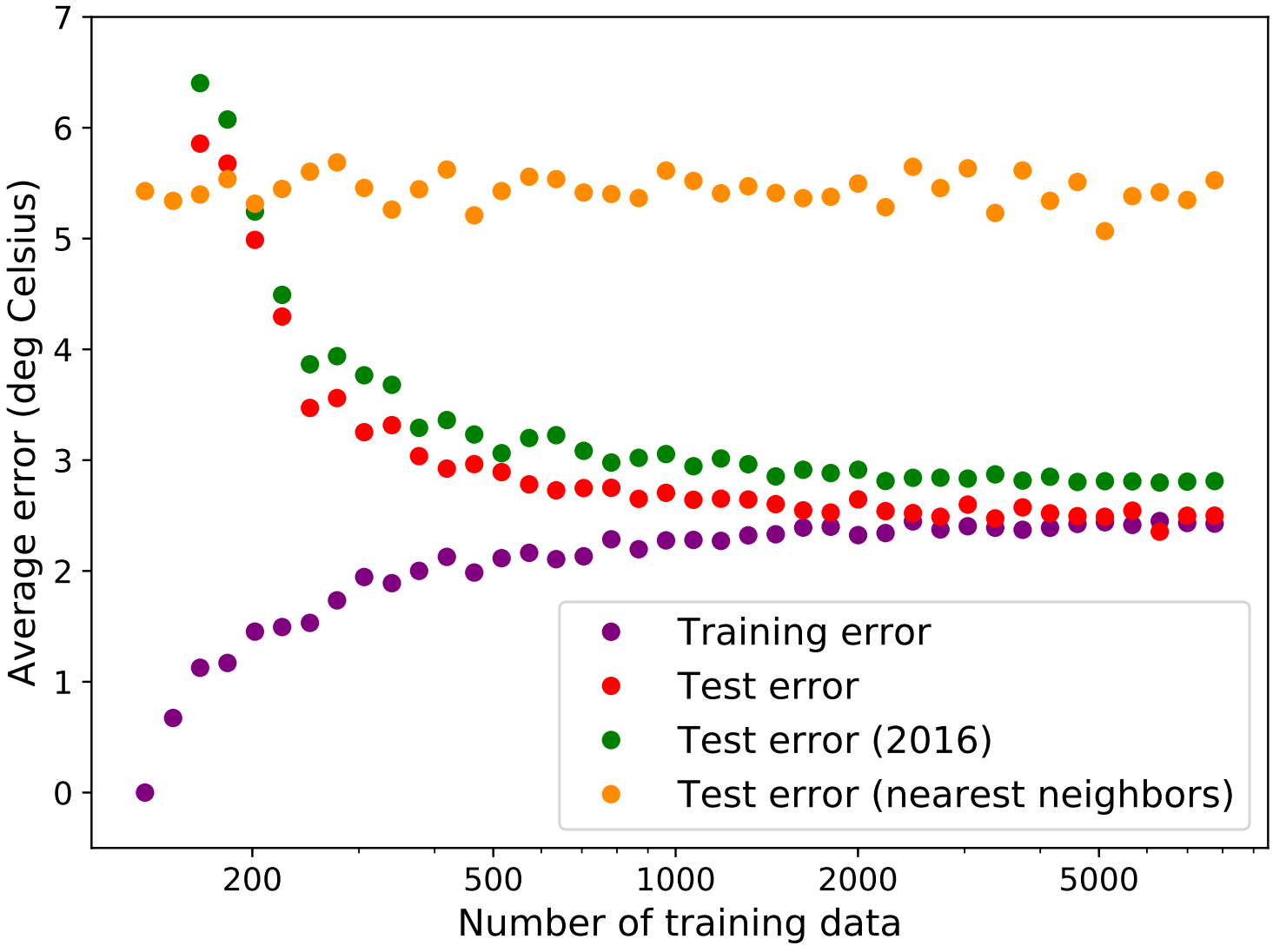
Strange claim

I found a cool way to predict the daily temperature in New York: It's just a linear combination of the temperature in every other state. I fit the model on data from the last month and a half and it's almost perfect!

Temperature prediction via linear regression

- Dataset of hourly temperatures measured at weather stations all over the US
- Goal: Predict temperature in Yosemite from other temperatures
- Response: Temperature in Yosemite
- Features: Temperatures in 133 other stations ($p = 133$) in 2015
- Test set: 10^3 measurements
- Additional test set: All measurements from 2016

Results


Average error (deg Celsius)
0
1
2
3
4
5
6
7
200
500
1000
2000
5000
Number of training data
Training error
Test error
Test error (2016)
Test error (nearest neighbors)

Linear model

To analyze the performance of the least-squares estimator we assume a linear model with additive noise

$$\vec{y}_{\text{train}} := X_{\text{train}} \vec{\beta}_{\text{true}} + \vec{z}_{\text{train}}$$

The LS estimator equals

$$\vec{\beta}_{\text{LS}} := \arg\min_{\vec{\beta}} \; \|\vec{y}_{\text{train}} - X_{\text{train}} \vec{\beta}\|_2$$

Noise model

We assume that $\vec{z}_{\text{train}}$ is iid Gaussian with zero mean and variance σ^2

Under this assumption $\vec{\beta}_{\text{LS}}$ is the maximum-likelihood estimate

Maximum likelihood estimate

Likelihood of iid Gaussian samples with mean $X\vec{\beta}$ and variance σ^2

$$\mathcal{L}_{\vec{y}}(\vec{\beta}) = \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \exp\left(-\frac{1}{2\sigma^2}\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2\right)$$

Maximum likelihood estimate

Likelihood of iid Gaussian samples with mean $X\vec{\beta}$ and variance σ^2

$$\mathcal{L}_{\vec{y}}(\vec{\beta}) = \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \exp\left(-\frac{1}{2\sigma^2}\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2\right)$$

To find the ML estimate, we maximize the log likelihood

$$\begin{aligned}
\vec{\beta}_{\mathsf{ML}} &= \arg\max_{\vec{\beta}} \mathcal{L}_{\vec{y}}\left(\vec{\beta}\right) \\
&= \arg\max_{\vec{\beta}} \log \mathcal{L}_{\vec{y}}\left(\vec{\beta}\right) \\
&= \arg\min_{\vec{\beta}} \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2
\end{aligned}$$

Coefficient error

If the training data follow the linear model and X_{train} is full rank,

$$\vec{\beta}_{\text{LS}} - \vec{\beta}_{\text{true}} = \left(X_{\text{train}}^T X_{\text{train}}\right)^{-1} X_{\text{train}}^T \vec{y}_{\text{train}} - \vec{\beta}_{\text{true}}$$

Coefficient error

If the training data follow the linear model and X_{train} is full rank,

$$\begin{aligned}\vec{\beta}_{\text{LS}} - \vec{\beta}_{\text{true}} &= \left(X_{\text{train}}^T X_{\text{train}}\right)^{-1} X_{\text{train}}^T \vec{y}_{\text{train}} - \vec{\beta}_{\text{true}} \\ &= \left(X_{\text{train}}^T X_{\text{train}}\right)^{-1} X_{\text{train}}^T \left(X_{\text{train}} \vec{\beta}_{\text{true}} + \vec{z}_{\text{train}}\right) - \vec{\beta}_{\text{true}}\end{aligned}$$

Coefficient error

If the training data follow the linear model and X_{train} is full rank,

$$\begin{aligned}
\vec{\beta}_{\text{LS}} - \vec{\beta}_{\text{true}} &= \left(X_{\text{train}}^T X_{\text{train}}\right)^{-1} X_{\text{train}}^T \vec{y}_{\text{train}} - \vec{\beta}_{\text{true}} \\
&= \left(X_{\text{train}}^T X_{\text{train}}\right)^{-1} X_{\text{train}}^T \left(X_{\text{train}} \vec{\beta}_{\text{true}} + \vec{z}_{\text{train}}\right) - \vec{\beta}_{\text{true}} \\
&= \vec{\beta}_{\text{true}} + \left(X_{\text{train}}^T X_{\text{train}}\right)^{-1} X_{\text{train}}^T \vec{z}_{\text{train}} - \vec{\beta}_{\text{true}}
\end{aligned}$$

Coefficient error

If the training data follow the linear model and X_{train} is full rank,

$$\begin{aligned}
\vec{\beta}_{\text{LS}} - \vec{\beta}_{\text{true}} &= \left(X_{\text{train}}^T X_{\text{train}}\right)^{-1} X_{\text{train}}^T \vec{y}_{\text{train}} - \vec{\beta}_{\text{true}} \\
&= \left(X_{\text{train}}^T X_{\text{train}}\right)^{-1} X_{\text{train}}^T \left(X_{\text{train}} \vec{\beta}_{\text{true}} + \vec{z}_{\text{train}}\right) - \vec{\beta}_{\text{true}} \\
&= \vec{\beta}_{\text{true}} + \left(X_{\text{train}}^T X_{\text{train}}\right)^{-1} X_{\text{train}}^T \vec{z}_{\text{train}} - \vec{\beta}_{\text{true}} \\
&= \left(X_{\text{train}}^T X_{\text{train}}\right)^{-1} X_{\text{train}}^T \vec{z}_{\text{train}}
\end{aligned}$$

Training error

The training error is the projection of the noise onto the orthogonal complement of the column space of X_{train}

$$\vec{y}_{\text{train}} - \vec{y}_{\text{LS}} = \vec{y}_{\text{train}} - \mathcal{P}_{\text{col}(X_{\text{train}})}\, \vec{y}_{\text{train}}$$

Training error

The training error is the projection of the noise onto the orthogonal complement of the column space of X_{train}

$$\begin{aligned}\vec{y}_{\text{train}} - \vec{y}_{\text{LS}} &= \vec{y}_{\text{train}} - \mathcal{P}_{\text{col}(X_{\text{train}})}\, \vec{y}_{\text{train}} \\ &= X_{\text{train}}\vec{\beta}_{\text{true}} + \vec{z}_{\text{train}} - \mathcal{P}_{\text{col}(X_{\text{train}})}\, (X_{\text{train}}\vec{\beta}_{\text{true}} + \vec{z}_{\text{train}})\end{aligned}$$

Training error

The training error is the projection of the noise onto the orthogonal complement of the column space of X_{train}

$$\begin{aligned}
\vec{y}_{\text{train}} - \vec{y}_{\text{LS}} &= \vec{y}_{\text{train}} - \mathcal{P}_{\text{col}(X_{\text{train}})}\, \vec{y}_{\text{train}} \\
&= X_{\text{train}}\vec{\beta}_{\text{true}} + \vec{z}_{\text{train}} - \mathcal{P}_{\text{col}(X_{\text{train}})}\, (X_{\text{train}}\vec{\beta}_{\text{true}} + \vec{z}_{\text{train}}) \\
&= X_{\text{train}}\vec{\beta}_{\text{true}} + \vec{z}_{\text{train}} - X_{\text{train}}\vec{\beta}_{\text{true}} - \mathcal{P}_{\text{col}(X_{\text{train}})}\, \vec{z}_{\text{train}}
\end{aligned}$$

Training error

The training error is the projection of the noise onto the orthogonal complement of the column space of X_{train}

$$\begin{aligned}
\vec{y}_{\text{train}} - \vec{y}_{\text{LS}} &= \vec{y}_{\text{train}} - \mathcal{P}_{\text{col}(X_{\text{train}})}\, \vec{y}_{\text{train}} \\
&= X_{\text{train}}\vec{\beta}_{\text{true}} + \vec{z}_{\text{train}} - \mathcal{P}_{\text{col}(X_{\text{train}})}\, (X_{\text{train}}\vec{\beta}_{\text{true}} + \vec{z}_{\text{train}}) \\
&= X_{\text{train}}\vec{\beta}_{\text{true}} + \vec{z}_{\text{train}} - X_{\text{train}}\vec{\beta}_{\text{true}} - \mathcal{P}_{\text{col}(X_{\text{train}})}\, \vec{z}_{\text{train}} \\
&= \mathcal{P}_{\text{col}(X_{\text{train}})^{\perp}}\, \vec{z}_{\text{train}}
\end{aligned}$$

Concentration of projection of Gaussian vector

Let $\mathcal{S}$ be a k-dimensional subspace of $\mathbb{R}^n$ and $\vec{\mathbf{z}} \in \mathbb{R}^n$ a vector of iid Gaussian noise with variance σ^2. For any $\epsilon \in (0, 1)$

$$\sigma\sqrt{k\left(1-\epsilon\right)} \leq ||\mathcal{P}_{\mathcal{S}}\,\vec{\mathbf{z}}||_2 \leq \sigma\sqrt{k\left(1+\epsilon\right)}$$

with probability at least $1 - 2\exp\left(-k\epsilon^2/8\right)$

Training error

Dimension of orthogonal complement of $\text{col}(X_{\text{train}})$ equals $n - p$

$$\text{Training RMSE} := \sqrt{\frac{||\vec{y}_{\text{train}} - \vec{y}_{\text{LS}}||_2^2}{n}}$$

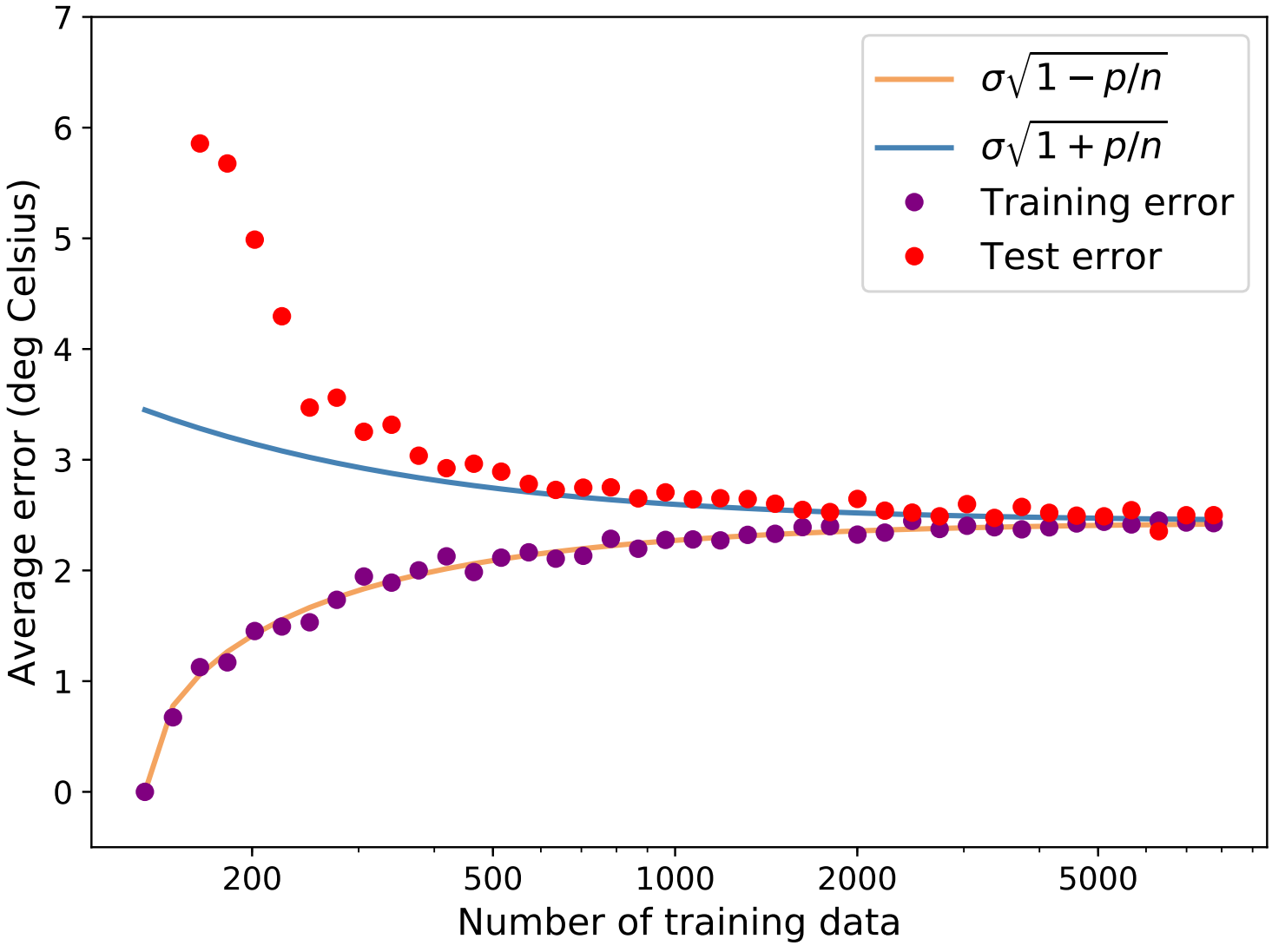
Training error

Dimension of orthogonal complement of $\text{col}(X_{\text{train}})$ equals $n - p$

$$\text{Training RMSE} := \sqrt{\frac{||\vec{y}_{\text{train}} - \vec{y}_{\text{LS}}||_2^2}{n}}$$

$$\approx \sigma \sqrt{1 - \frac{p}{n}}$$

Temperature prediction via linear regression



Overfitting

When $p \approx n$, the training error is very low!

Lower than error achieved by true coefficients

$$\text{Ideal training RMSE} := \sqrt{\frac{\left|\left|\vec{y}_{\text{train}} - X_{\text{train}}\vec{\beta}_{\text{true}}\right|\right|_2^2}{n}}$$

$$= \sqrt{\frac{\left|\left|\vec{z}_{\text{train}}\right|\right|_2^2}{n}} \approx \sigma$$

This indicates overfitting of noise

Test error

What we really care about is performance on held-out data

$$y_{\text{test}} := \langle \vec{x}_{\text{test}}, \vec{\beta}_{\text{true}} \rangle + z_{\text{test}}$$

The least-squares estimate equals

$$y_{\text{LS}} := \langle \vec{x}_{\text{test}}, \vec{\beta}_{\text{LS}} \rangle$$

where $\vec{\beta}_{\text{LS}}$ is computed from the training data

Model

Training and test noise are iid Gaussian with variance σ^2

Test feature vector $\vec{\mathbf{x}}_{\text{test}}$ is a zero-mean random vector

Training and test noise, and test features are all independent

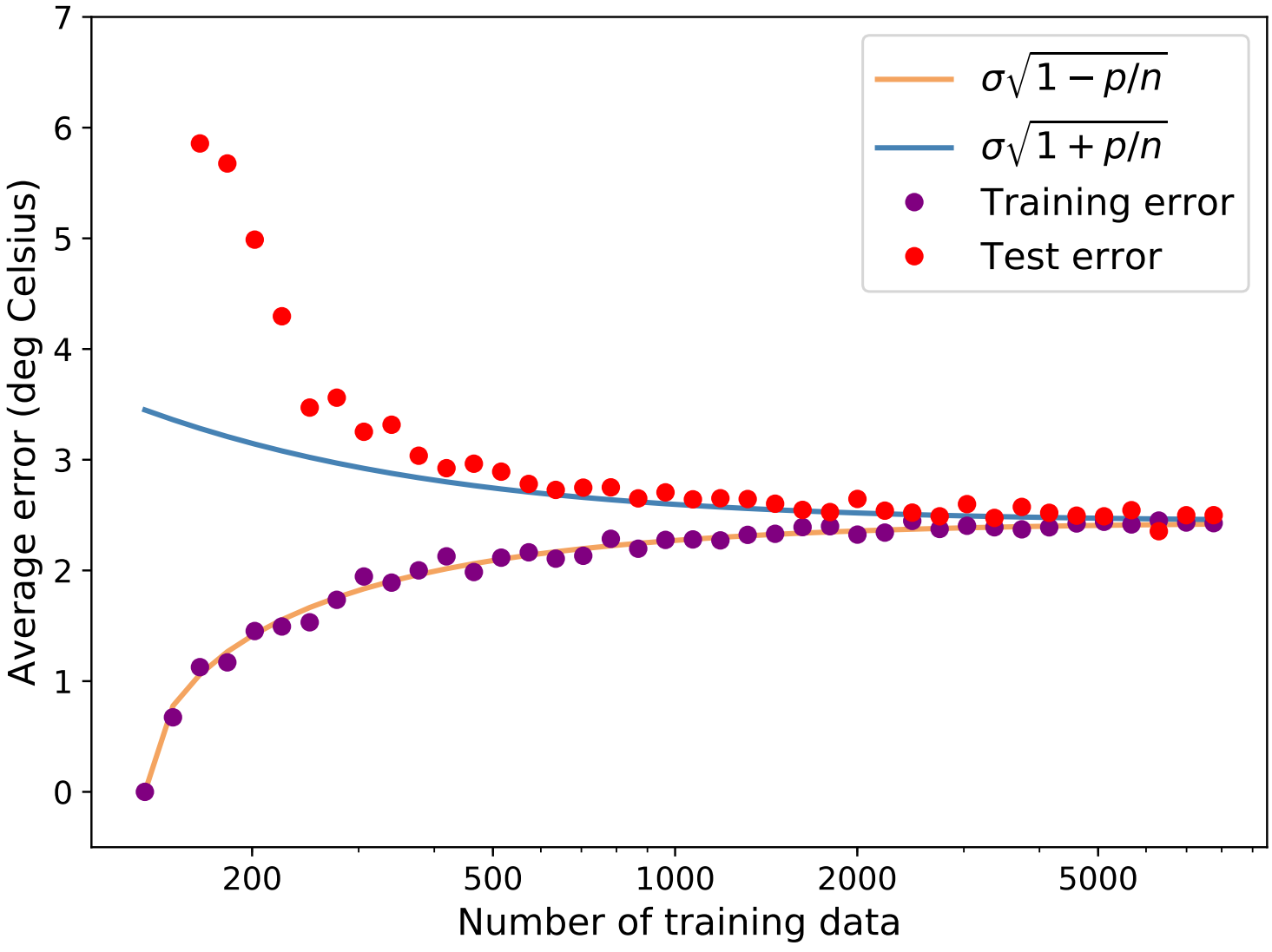
Test error

$$\text{Test RMSE} := \sqrt{\mathrm{E}\left((\boldsymbol{y}_{\text{test}} - \boldsymbol{y}_{\text{LS}})^2\right)} \approx \sigma\sqrt{1 + \frac{p}{n}}$$

as long as

$$\mathrm{E}\left(\vec{\mathbf{x}}_{\text{test}}\vec{\mathbf{x}}_{\text{test}}^{T}\right) \approx \frac{1}{n}X_{\text{train}}^{T}X_{\text{train}}$$

Temperature prediction via linear regression



Proof

$$\boldsymbol{y}_{\mathsf{test}} - \boldsymbol{y}_{\mathsf{LS}} = \langle \vec{\mathbf{x}}_{\mathsf{test}}, \vec{\beta}_{\mathsf{true}} \rangle + \boldsymbol{z}_{\mathsf{test}} - \langle \vec{\mathbf{x}}_{\mathsf{test}}, \vec{\boldsymbol{\beta}}_{\mathsf{LS}} \rangle$$

Proof

$$\begin{aligned} \boldsymbol{y}_{\text{test}} - \boldsymbol{y}_{\text{LS}} &= \langle \vec{\mathbf{x}}_{\text{test}}, \vec{\beta}_{\text{true}} \rangle + \boldsymbol{z}_{\text{test}} - \langle \vec{\mathbf{x}}_{\text{test}}, \vec{\boldsymbol{\beta}}_{\text{LS}} \rangle \\ &= \langle \vec{\mathbf{x}}_{\text{test}}, \vec{\beta}_{\text{true}} - \vec{\boldsymbol{\beta}}_{\text{LS}} \rangle + \boldsymbol{z}_{\text{test}} \end{aligned}$$

Proof

$$
\begin{aligned}
\boldsymbol{y}_{\text{test}} - \boldsymbol{y}_{\text{LS}} &= \langle \vec{\mathbf{x}}_{\text{test}}, \vec{\beta}_{\text{true}} \rangle + \boldsymbol{z}_{\text{test}} - \langle \vec{\mathbf{x}}_{\text{test}}, \vec{\boldsymbol{\beta}}_{\text{LS}} \rangle \\
&= \langle \vec{\mathbf{x}}_{\text{test}}, \vec{\beta}_{\text{true}} - \vec{\boldsymbol{\beta}}_{\text{LS}} \rangle + \boldsymbol{z}_{\text{test}} \\
&= -\langle \vec{\mathbf{x}}_{\text{test}}, \left(X_{\text{train}}^{T} X_{\text{train}} \right)^{-1} X_{\text{train}}^{T} \vec{\boldsymbol{z}}_{\text{train}} \rangle + \boldsymbol{z}_{\text{test}}
\end{aligned}
$$

Proof

$$
\begin{aligned}
\boldsymbol{y}_{\text{test}} - \boldsymbol{y}_{\text{LS}} &= \langle \vec{\mathbf{x}}_{\text{test}}, \vec{\beta}_{\text{true}} \rangle + \boldsymbol{z}_{\text{test}} - \langle \vec{\mathbf{x}}_{\text{test}}, \vec{\boldsymbol{\beta}}_{\text{LS}} \rangle \\
&= \langle \vec{\mathbf{x}}_{\text{test}}, \vec{\beta}_{\text{true}} - \vec{\boldsymbol{\beta}}_{\text{LS}} \rangle + \boldsymbol{z}_{\text{test}} \\
&= -\langle \vec{\mathbf{x}}_{\text{test}}, \left(X_{\text{train}}^{T} X_{\text{train}} \right)^{-1} X_{\text{train}}^{T} \vec{\boldsymbol{z}}_{\text{train}} \rangle + \boldsymbol{z}_{\text{test}} \\
&= -\langle \vec{\mathbf{x}}_{\text{test}}, X^{\dagger} \vec{\boldsymbol{z}}_{\text{train}} \rangle + \boldsymbol{z}_{\text{test}}
\end{aligned}
$$

Proof

$$\mathrm{E}\left((\boldsymbol{y}_{\mathsf{test}} - \boldsymbol{y}_{\mathsf{LS}})^2\right) = \mathrm{E}\left(\langle \vec{\mathbf{x}}_{\mathsf{test}}, X^{\dagger}\vec{\boldsymbol{z}}_{\mathsf{train}}\rangle^2\right) + \mathrm{E}\left(\boldsymbol{z}_{\mathsf{test}}^2\right)$$

Proof

$$\begin{aligned}
\mathrm{E}\left((\boldsymbol{y}_{\mathsf{test}} - \boldsymbol{y}_{\mathsf{LS}})^2\right) &= \mathrm{E}\left(\langle \vec{\mathbf{x}}_{\mathsf{test}}, X^{\dagger}\vec{\boldsymbol{z}}_{\mathsf{train}}\rangle^2\right) + \mathrm{E}\left(\boldsymbol{z}_{\mathsf{test}}^2\right) \\
&= \mathrm{E}\left(\vec{\mathbf{x}}_{\mathsf{test}}^{T} X^{\dagger}\vec{\boldsymbol{z}}_{\mathsf{train}}\vec{\boldsymbol{z}}_{\mathsf{train}}^{T}(X^{\dagger})^{T}\vec{\mathbf{x}}_{\mathsf{test}}\right) + \sigma^2
\end{aligned}$$

Proof

$$
\begin{aligned}
\mathrm{E}\left((\boldsymbol{y}_{\mathsf{test}} - \boldsymbol{y}_{\mathsf{LS}})^2\right) &= \mathrm{E}\left(\langle \vec{\mathbf{x}}_{\mathsf{test}}, X^{\dagger}\vec{\boldsymbol{z}}_{\mathsf{train}}\rangle^2\right) + \mathrm{E}\left(\boldsymbol{z}_{\mathsf{test}}^2\right) \\
&= \mathrm{E}\left(\vec{\mathbf{x}}_{\mathsf{test}}^{T} X^{\dagger}\vec{\boldsymbol{z}}_{\mathsf{train}}\vec{\boldsymbol{z}}_{\mathsf{train}}^{T}(X^{\dagger})^{T}\vec{\mathbf{x}}_{\mathsf{test}}\right) + \sigma^2 \\
&= \mathrm{E}\left(\mathrm{E}\left(\vec{\mathbf{x}}_{\mathsf{test}}^{T} X^{\dagger}\vec{\boldsymbol{z}}_{\mathsf{train}}\vec{\boldsymbol{z}}_{\mathsf{train}}^{T}(X^{\dagger})^{T}\vec{\mathbf{x}}_{\mathsf{test}} \mid \vec{\mathbf{x}}_{\mathsf{test}}\right)\right) + \sigma^2
\end{aligned}
$$

Proof

$$
\begin{aligned}
\mathrm{E}\left((\boldsymbol{y}_{\mathsf{test}} - \boldsymbol{y}_{\mathsf{LS}})^2\right) &= \mathrm{E}\left(\langle \vec{\mathbf{x}}_{\mathsf{test}}, X^{\dagger}\vec{\boldsymbol{z}}_{\mathsf{train}}\rangle^2\right) + \mathrm{E}\left(\boldsymbol{z}_{\mathsf{test}}^2\right) \\
&= \mathrm{E}\left(\vec{\mathbf{x}}_{\mathsf{test}}^{T} X^{\dagger}\vec{\boldsymbol{z}}_{\mathsf{train}}\vec{\boldsymbol{z}}_{\mathsf{train}}^{T}(X^{\dagger})^{T}\vec{\mathbf{x}}_{\mathsf{test}}\right) + \sigma^2 \\
&= \mathrm{E}\left(\mathrm{E}\left(\vec{\mathbf{x}}_{\mathsf{test}}^{T} X^{\dagger}\vec{\boldsymbol{z}}_{\mathsf{train}}\vec{\boldsymbol{z}}_{\mathsf{train}}^{T}(X^{\dagger})^{T}\vec{\mathbf{x}}_{\mathsf{test}} \mid \vec{\mathbf{x}}_{\mathsf{test}}\right)\right) + \sigma^2 \\
&= \mathrm{E}\left(\vec{\mathbf{x}}_{\mathsf{test}}^{T} X^{\dagger}\mathrm{E}\left(\vec{\boldsymbol{z}}_{\mathsf{train}}\vec{\boldsymbol{z}}_{\mathsf{train}}^{T}\right)(X^{\dagger})^{T}\vec{\mathbf{x}}_{\mathsf{test}}\right) + \sigma^2
\end{aligned}
$$

Proof

$$
\begin{aligned}
\mathrm{E}\left((\boldsymbol{y}_{\mathsf{test}} - \boldsymbol{y}_{\mathsf{LS}})^2\right) &= \mathrm{E}\left(\langle \vec{\mathbf{x}}_{\mathsf{test}}, X^{\dagger}\vec{\boldsymbol{z}}_{\mathsf{train}}\rangle^2\right) + \mathrm{E}\left(\boldsymbol{z}_{\mathsf{test}}^2\right) \\
&= \mathrm{E}\left(\vec{\mathbf{x}}_{\mathsf{test}}^T X^{\dagger}\vec{\boldsymbol{z}}_{\mathsf{train}}\vec{\boldsymbol{z}}_{\mathsf{train}}^T (X^{\dagger})^T \vec{\mathbf{x}}_{\mathsf{test}}\right) + \sigma^2 \\
&= \mathrm{E}\left(\mathrm{E}\left(\vec{\mathbf{x}}_{\mathsf{test}}^T X^{\dagger}\vec{\boldsymbol{z}}_{\mathsf{train}}\vec{\boldsymbol{z}}_{\mathsf{train}}^T (X^{\dagger})^T \vec{\mathbf{x}}_{\mathsf{test}} \mid \vec{\mathbf{x}}_{\mathsf{test}}\right)\right) + \sigma^2 \\
&= \mathrm{E}\left(\vec{\mathbf{x}}_{\mathsf{test}}^T X^{\dagger}\mathrm{E}\left(\vec{\boldsymbol{z}}_{\mathsf{train}}\vec{\boldsymbol{z}}_{\mathsf{train}}^T\right)(X^{\dagger})^T \vec{\mathbf{x}}_{\mathsf{test}}\right) + \sigma^2 \\
&= \sigma^2\,\mathrm{E}\left(\vec{\mathbf{x}}_{\mathsf{test}}^T X^{\dagger}(X^{\dagger})^T \vec{\mathbf{x}}_{\mathsf{test}}\right) + \sigma^2
\end{aligned}
$$

Proof

$$
\begin{aligned}
X^{\dagger}(X^{\dagger})^{T} &= \left(X_{\text{train}}^{T} X_{\text{train}}\right)^{-1} X_{\text{train}}^{T} X_{\text{train}} \left(X_{\text{train}}^{T} X_{\text{train}}\right)^{-1} \\
&= \left(X_{\text{train}}^{T} X_{\text{train}}\right)^{-1}
\end{aligned}
$$

Proof

$$
\begin{aligned}
X^{\dagger}(X^{\dagger})^{T} &= \left(X_{\text{train}}^{T} X_{\text{train}}\right)^{-1} X_{\text{train}}^{T} X_{\text{train}} \left(X_{\text{train}}^{T} X_{\text{train}}\right)^{-1} \\
&= \left(X_{\text{train}}^{T} X_{\text{train}}\right)^{-1}
\end{aligned}
$$

$$
\mathrm{E}\left((\boldsymbol{y}_{\text{test}} - \boldsymbol{y}_{\text{LS}})^{2}\right) = \sigma^{2}\,\mathrm{E}\left(\vec{\mathbf{x}}_{\text{test}}^{T}\left(X_{\text{train}}^{T} X_{\text{train}}\right)^{-1}\vec{\mathbf{x}}_{\text{test}}\right) + \sigma^{2}
$$

Proof

$$\mathrm{E}\left(\vec{\mathsf{x}}_{\mathsf{test}}^{T}\left(X_{\mathsf{train}}^{T}X_{\mathsf{train}}\right)^{-1}\vec{\mathsf{x}}_{\mathsf{test}}\right)=\mathrm{E}\left(\mathsf{tr}\left(\vec{\mathsf{x}}_{\mathsf{test}}^{T}\left(X_{\mathsf{train}}^{T}X_{\mathsf{train}}\right)^{-1}\vec{\mathsf{x}}_{\mathsf{test}}\right)\right)$$

Proof

$$
\begin{aligned}
\mathrm{E}\left(\vec{\mathbf{x}}_{\mathsf{test}}^{T}\left(X_{\mathsf{train}}^{T}X_{\mathsf{train}}\right)^{-1}\vec{\mathbf{x}}_{\mathsf{test}}\right) &= \mathrm{E}\left(\mathsf{tr}\left(\vec{\mathbf{x}}_{\mathsf{test}}^{T}\left(X_{\mathsf{train}}^{T}X_{\mathsf{train}}\right)^{-1}\vec{\mathbf{x}}_{\mathsf{test}}\right)\right) \\
&= \mathrm{E}\left(\mathsf{tr}\left(\left(X_{\mathsf{train}}^{T}X_{\mathsf{train}}\right)^{-1}\vec{\mathbf{x}}_{\mathsf{test}}\vec{\mathbf{x}}_{\mathsf{test}}^{T}\right)\right) \\
&= \mathsf{tr}\left(\left(X_{\mathsf{train}}^{T}X_{\mathsf{train}}\right)^{-1}\mathrm{E}\left(\vec{\mathbf{x}}_{\mathsf{test}}\vec{\mathbf{x}}_{\mathsf{test}}^{T}\right)\right) \\
&\approx \frac{1}{n}\,\mathsf{tr}\left(I\right) = \frac{p}{n}
\end{aligned}
$$

SVD analysis

$$
\begin{aligned}
\vec{\beta}_{\mathsf{LS}} - \vec{\beta}_{\mathsf{true}} &= \left(X_{\mathsf{train}}^T X_{\mathsf{train}} \right)^{-1} X_{\mathsf{train}}^T \vec{z}_{\mathsf{train}} \\
&= V_{\mathsf{train}} S_{\mathsf{train}}^{-1} U_{\mathsf{train}}^T \vec{z}_{\mathsf{train}} \\
&= \sum_{i=1}^{p} \frac{\left\langle \vec{u}_i, \vec{z}_{\mathsf{train}} \right\rangle}{s_i} \vec{v}_i
\end{aligned}
$$

SVD analysis

$$\begin{aligned}\vec{\beta}_{\mathsf{LS}} - \vec{\beta}_{\mathsf{true}} &= \left(X_{\mathsf{train}}^T X_{\mathsf{train}}\right)^{-1} X_{\mathsf{train}}^T \vec{z}_{\mathsf{train}} \\ &= V_{\mathsf{train}} S_{\mathsf{train}}^{-1} U_{\mathsf{train}}^T \vec{z}_{\mathsf{train}} \\ &= \sum_{i=1}^{p} \frac{\langle \vec{u}_i, \vec{z}_{\mathsf{train}} \rangle}{s_i} \vec{v}_i\end{aligned}$$

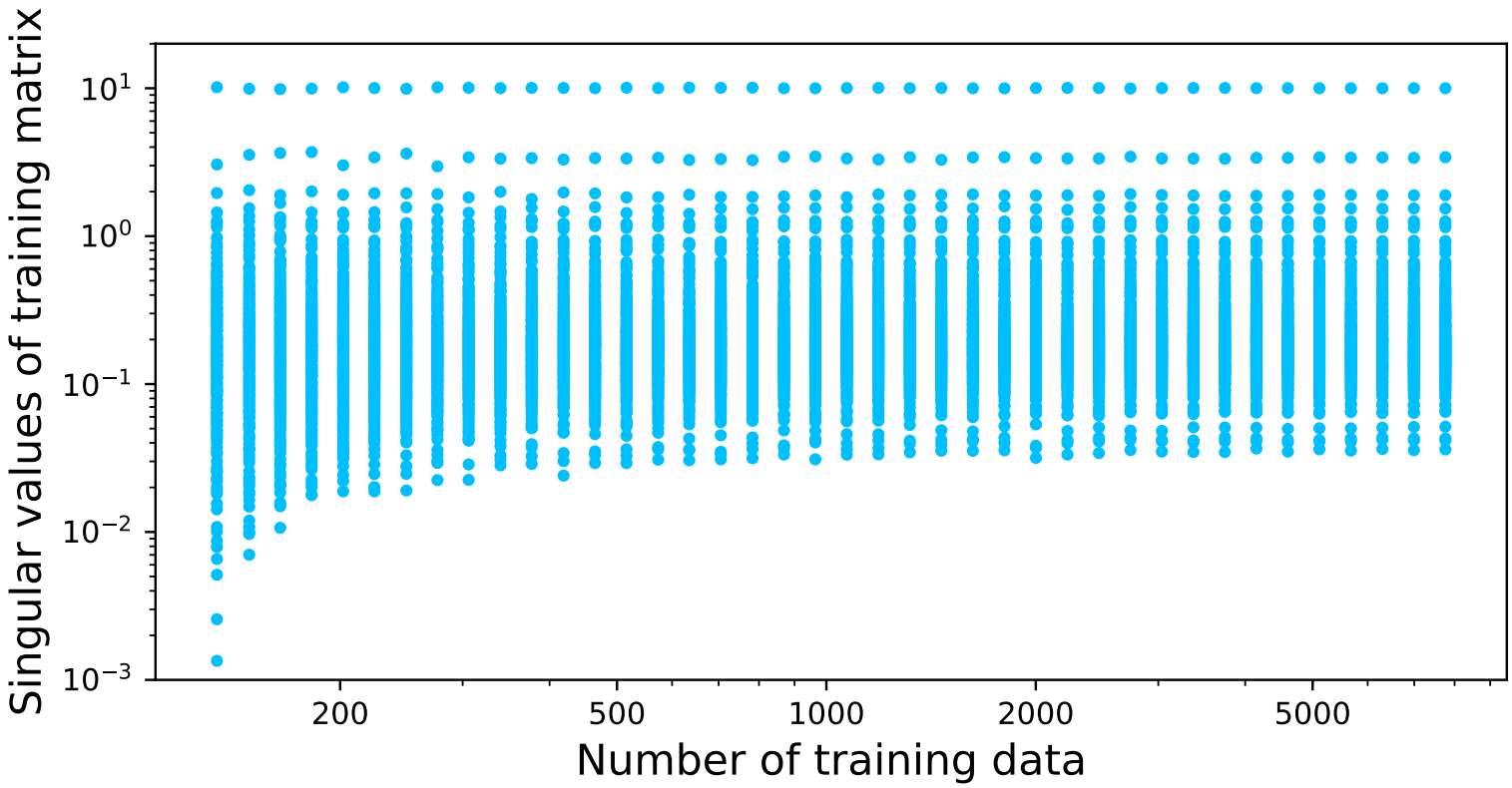
$$y_{\mathsf{test}} - y_{\mathsf{LS}} = \langle \vec{x}_{\mathsf{test}}, \vec{\beta}_{\mathsf{true}} - \vec{\beta}_{\mathsf{LS}} \rangle + z_{\mathsf{test}}$$

SVD analysis

$$\begin{aligned}
\vec{\beta}_{\mathsf{LS}} - \vec{\beta}_{\mathsf{true}} &= \left(X_{\mathsf{train}}^T X_{\mathsf{train}}\right)^{-1} X_{\mathsf{train}}^T \vec{z}_{\mathsf{train}} \\
&= V_{\mathsf{train}} S_{\mathsf{train}}^{-1} U_{\mathsf{train}}^T \vec{z}_{\mathsf{train}} \\
&= \sum_{i=1}^{p} \frac{\left\langle \vec{u}_i, \vec{z}_{\mathsf{train}} \right\rangle}{s_i} \vec{v}_i
\end{aligned}$$

$$\begin{aligned}
y_{\mathsf{test}} - y_{\mathsf{LS}} &= \left\langle \vec{x}_{\mathsf{test}}, \vec{\beta}_{\mathsf{true}} - \vec{\beta}_{\mathsf{LS}} \right\rangle + z_{\mathsf{test}} \\
&= z_{\mathsf{test}} - \sum_{i=1}^{p} \frac{\left\langle \vec{x}_{\mathsf{test}}, \vec{v}_i \right\rangle \left\langle \vec{u}_i, \vec{z}_{\mathsf{train}} \right\rangle}{s_i}
\end{aligned}$$

Temperature prediction via linear regression



SVD analysis

If sample covariance of training matrix is close to true covariance matrix

$$\mathrm{E}\left(\langle \vec{\mathbf{x}}_{\text{test}}, \vec{v}_i \rangle^2\right) = \mathrm{E}\left(\vec{v}_i{}^T \vec{\mathbf{x}}_{\text{test}} \vec{\mathbf{x}}_{\text{test}}^T \vec{v}_i\right)$$

SVD analysis

If sample covariance of training matrix is close to true covariance matrix

$$\begin{aligned} \mathrm{E}\left(\langle \vec{\mathbf{x}}_{\text{test}}, \vec{v}_i \rangle^2\right) &= \mathrm{E}\left(\vec{v}_i{}^T \vec{\mathbf{x}}_{\text{test}} \vec{\mathbf{x}}_{\text{test}}^T \vec{v}_i\right) \\ &= \vec{v}_i{}^T \mathrm{E}\left(\vec{\mathbf{x}}_{\text{test}} \vec{\mathbf{x}}_{\text{test}}^T\right) \vec{v}_i \end{aligned}$$

SVD analysis

If sample covariance of training matrix is close to true covariance matrix

$$\begin{aligned}
\mathrm{E}\left(\langle \vec{\mathbf{x}}_{\text{test}}, \vec{v}_i \rangle^2\right) &= \mathrm{E}\left(\vec{v}_i{}^T \vec{\mathbf{x}}_{\text{test}} \vec{\mathbf{x}}_{\text{test}}^T \vec{v}_i\right) \\
&= \vec{v}_i{}^T \mathrm{E}\left(\vec{\mathbf{x}}_{\text{test}} \vec{\mathbf{x}}_{\text{test}}^T\right) \vec{v}_i \\
&\approx \frac{1}{n} \vec{v}_i{}^T X_{\text{train}}^T X_{\text{train}} \vec{v}_i
\end{aligned}$$

SVD analysis

If sample covariance of training matrix is close to true covariance matrix

$$\begin{aligned}
\mathrm{E}\left(\langle \vec{\mathbf{x}}_{\text{test}}, \vec{v}_i \rangle^2\right) &= \mathrm{E}\left(\vec{v}_i{}^T \vec{\mathbf{x}}_{\text{test}} \vec{\mathbf{x}}_{\text{test}}^T \vec{v}_i\right) \\
&= \vec{v}_i{}^T \mathrm{E}\left(\vec{\mathbf{x}}_{\text{test}} \vec{\mathbf{x}}_{\text{test}}^T\right) \vec{v}_i \\
&\approx \frac{1}{n} \vec{v}_i{}^T X_{\text{train}}^T X_{\text{train}} \vec{v}_i \\
&= \frac{1}{n} \vec{v}_i{}^T V S^2 V^T \vec{v}_i
\end{aligned}$$

SVD analysis

If sample covariance of training matrix is close to true covariance matrix

$$
\begin{aligned}
\mathrm{E}\left(\langle \vec{\mathbf{x}}_{\text{test}}, \vec{v}_i \rangle^2\right) &= \mathrm{E}\left(\vec{v}_i{}^T \vec{\mathbf{x}}_{\text{test}} \vec{\mathbf{x}}_{\text{test}}^T \vec{v}_i\right) \\
&= \vec{v}_i{}^T \mathrm{E}\left(\vec{\mathbf{x}}_{\text{test}} \vec{\mathbf{x}}_{\text{test}}^T\right) \vec{v}_i \\
&\approx \frac{1}{n} \vec{v}_i{}^T X_{\text{train}}^T X_{\text{train}} \vec{v}_i \\
&= \frac{1}{n} \vec{v}_i{}^T V S^2 V^T \vec{v}_i \\
&= \frac{s_i^2}{n}
\end{aligned}
$$

SVD analysis

If sample covariance of training matrix is close to true covariance matrix

$$\begin{aligned}
\mathrm{E}\left(\langle \vec{\mathbf{x}}_{\text{test}}, \vec{v}_i \rangle^2\right) &= \mathrm{E}\left(\vec{v}_i{}^T \vec{\mathbf{x}}_{\text{test}} \vec{\mathbf{x}}_{\text{test}}^T \vec{v}_i\right) \\
&= \vec{v}_i{}^T \mathrm{E}\left(\vec{\mathbf{x}}_{\text{test}} \vec{\mathbf{x}}_{\text{test}}^T\right) \vec{v}_i \\
&\approx \frac{1}{n} \vec{v}_i{}^T X_{\text{train}}^T X_{\text{train}} \vec{v}_i \\
&= \frac{1}{n} \vec{v}_i{}^T V S^2 V^T \vec{v}_i \\
&= \frac{s_i^2}{n}
\end{aligned}$$

Otherwise: noise amplification

Motivating applications

The singular-value decomposition

Principal component analysis

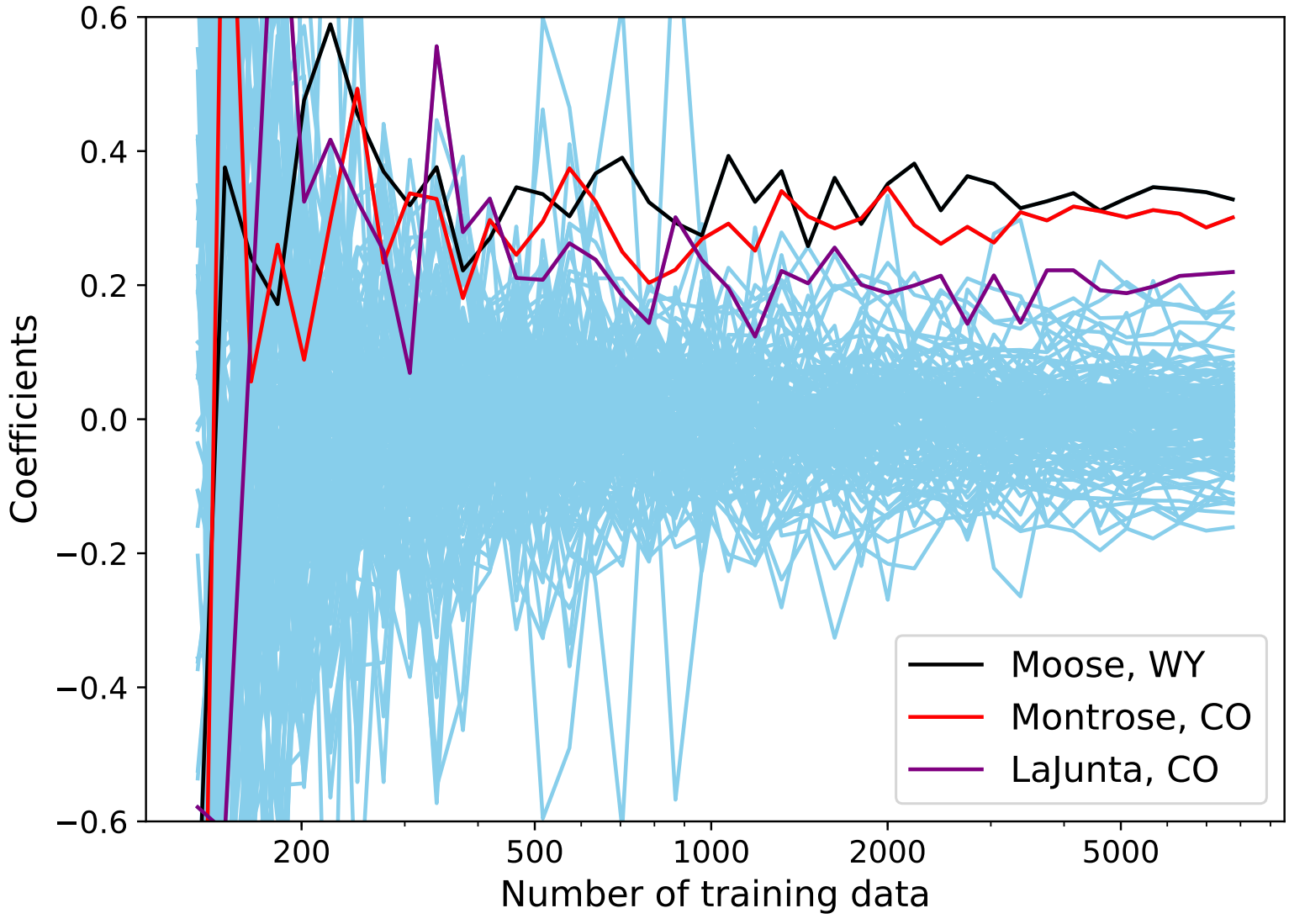
Dimensionality reduction

Linear regression

Ridge regression

Low-rank matrix estimation

Temperature prediction via linear regression



Motivation

Overfitting often reflected in large coefficients that cancel out to match the noise

Solution: Penalize large-norm solutions when fitting the model

Adding a penalty term to promote a particular structure is called regularization

Ridge regression

For a fixed regularization parameter $\lambda > 0$

$$\vec{\beta}_{\text{ridge}} := \arg\min_{\vec{\beta}} \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 + \lambda \left|\left|\vec{\beta}\right|\right|_2^2$$

Ridge regression

For a fixed regularization parameter $\lambda > 0$

$$\vec{\beta}_{\text{ridge}} := \arg\min_{\vec{\beta}} \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 + \lambda \left|\left|\vec{\beta}\right|\right|_2^2$$

$$= \left(X^T X + \lambda I\right)^{-1} X^T \vec{y}$$

Ridge regression

For a fixed regularization parameter $\lambda > 0$

$$\vec{\beta}_{\text{ridge}} := \arg\min_{\vec{\beta}} \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 + \lambda \left|\left|\vec{\beta}\right|\right|_2^2$$

$$= \left(X^T X + \lambda I\right)^{-1} X^T \vec{y}$$

When $\lambda \to 0$ then $\vec{\beta}_{\text{ridge}} \to \vec{\beta}_{\text{LS}}$

Ridge regression

For a fixed regularization parameter $\lambda > 0$

$$\vec{\beta}_{\text{ridge}} := \arg\min_{\vec{\beta}} \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 + \lambda \left|\left|\vec{\beta}\right|\right|_2^2$$

$$= \left(X^T X + \lambda I\right)^{-1} X^T \vec{y}$$

When $\lambda \to 0$ then $\vec{\beta}_{\text{ridge}} \to \vec{\beta}_{\text{LS}}$

When $\lambda \to \infty$ then $\vec{\beta}_{\text{ridge}} \to 0$

Proof

$\vec{\beta}_{\text{ridge}}$ is the solution to a modified least-squares problem

$$\vec{\beta}_{\text{ridge}} = \arg\min_{\vec{\beta}} \left\| \begin{bmatrix} \vec{y} \\ 0 \end{bmatrix} - \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix} \vec{\beta} \right\|_2^2$$

Proof

$\vec{\beta}_{\text{ridge}}$ is the solution to a modified least-squares problem

$$\begin{aligned}\vec{\beta}_{\text{ridge}} &= \arg\min_{\vec{\beta}} \left\| \begin{bmatrix} \vec{y} \\ 0 \end{bmatrix} - \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix} \vec{\beta} \right\|_2^2 \\ &= \left(\begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix}^T \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix} \right)^{-1} \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix}^T \begin{bmatrix} \vec{y} \\ 0 \end{bmatrix}\end{aligned}$$

Proof

$\vec{\beta}_{\text{ridge}}$ is the solution to a modified least-squares problem

$$\begin{aligned}
\vec{\beta}_{\text{ridge}} &= \arg\min_{\vec{\beta}} \left\| \begin{bmatrix} \vec{y} \\ 0 \end{bmatrix} - \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix} \vec{\beta} \right\|_2^2 \\
&= \left(\begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix}^T \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix} \right)^{-1} \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix}^T \begin{bmatrix} \vec{y} \\ 0 \end{bmatrix} \\
&= \left(X^T X + \lambda I \right)^{-1} X^T \vec{y}
\end{aligned}$$

Problem

How to calibrate regularization parameter

Cannot use coefficient error (we don't know the true value!)

Cannot minimize over training data (why?)

Solution: Check fit on new data

Cross validation

Given a set of examples

$$\left(y^{(1)}, \vec{x}^{(1)}\right), \left(y^{(2)}, \vec{x}^{(2)}\right), \ldots, \left(y^{(n)}, \vec{x}^{(n)}\right),$$

1. Partition data into a training set $X_{\text{train}} \in \mathbb{R}^{n_{\text{train}} \times p}$, $\vec{y}_{\text{train}} \in \mathbb{R}^{n_{\text{train}}}$ and a validation set $X_{\text{val}} \in \mathbb{R}^{n_{\text{val}} \times p}$, $\vec{y}_{\text{val}} \in \mathbb{R}^{n_{\text{val}}}$

2. Fit model using the training set for every λ in a set Λ

$$\vec{\beta}_{\text{ridge}}(\lambda) := \arg\min_{\vec{\beta}} \left|\left|\vec{y}_{\text{train}} - X_{\text{train}}\vec{\beta}\right|\right|_2^2 + \lambda \left|\left|\vec{\beta}\right|\right|_2^2$$

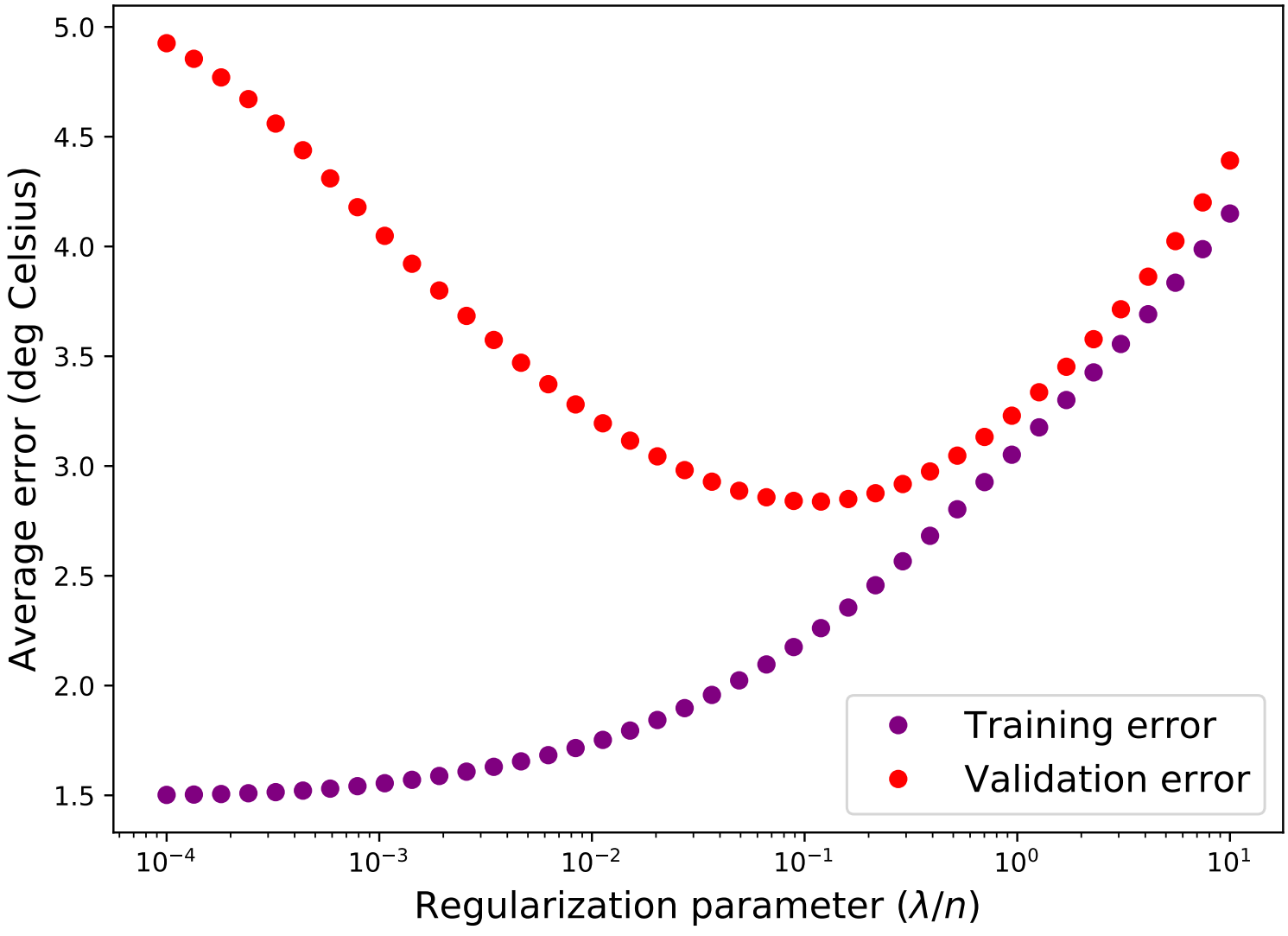
   and evaluate the fitting error on the validation set

$$\text{err}(\lambda) := \left|\left|\vec{y}_{\text{train}} - X_{\text{train}}\vec{\beta}_{\text{ridge}}(\lambda)\right|\right|_2^2$$

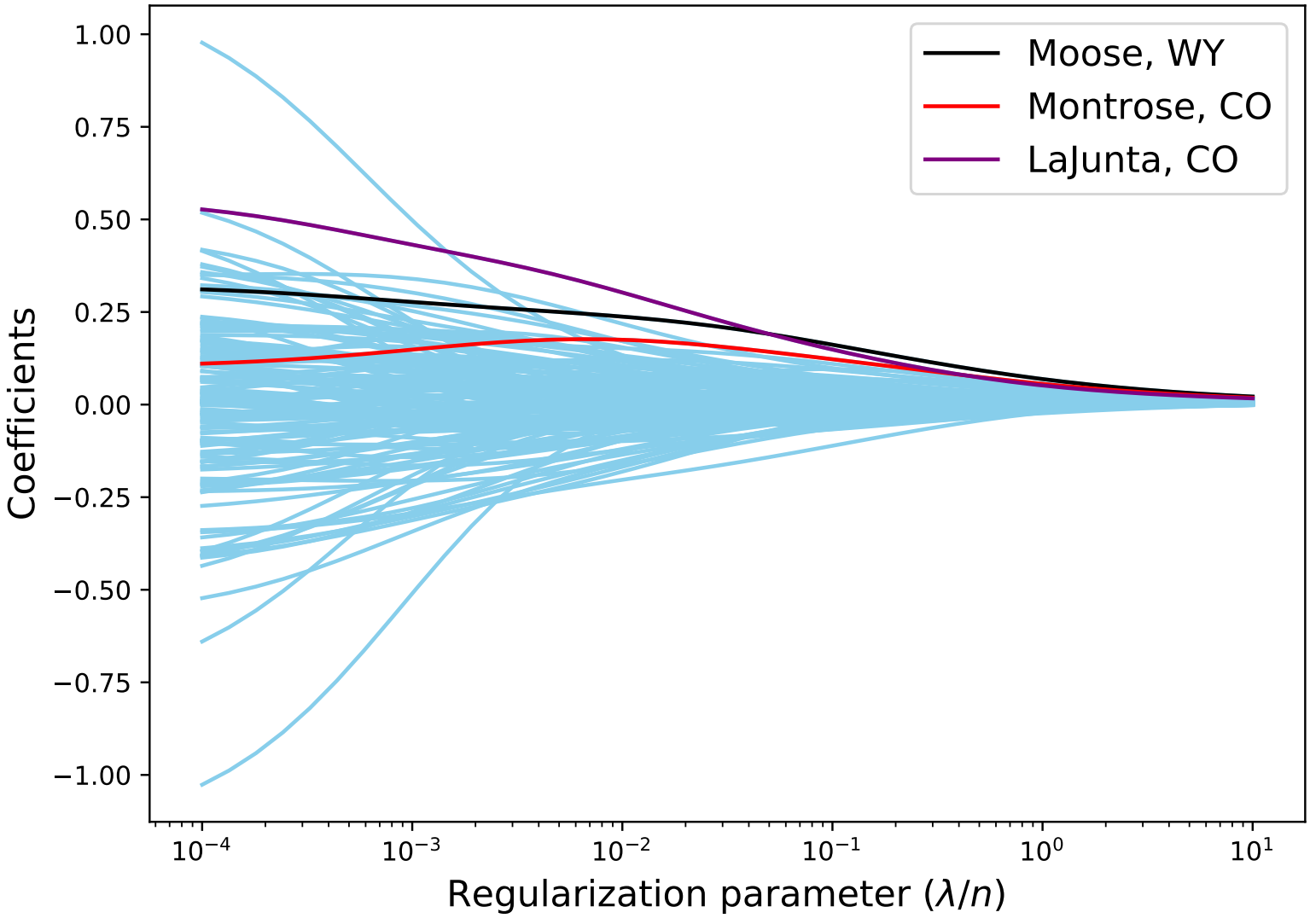
3. Choose the value of λ that minimizes the validation-set error

$$\lambda_{\text{cv}} := \arg\min_{\lambda \in \Lambda} \text{err}(\lambda)$$

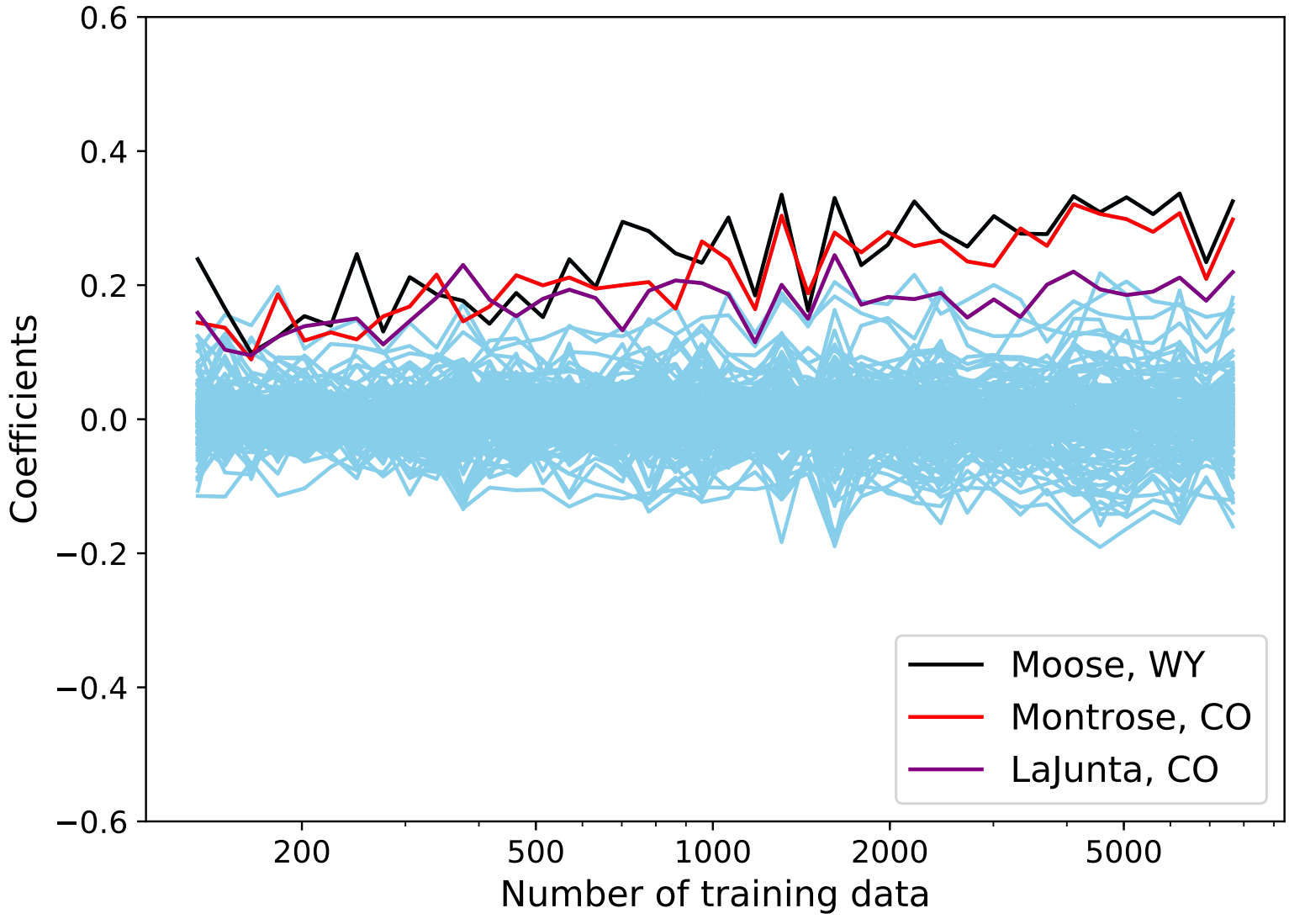
Temperature prediction via linear regression ($n = 202$)



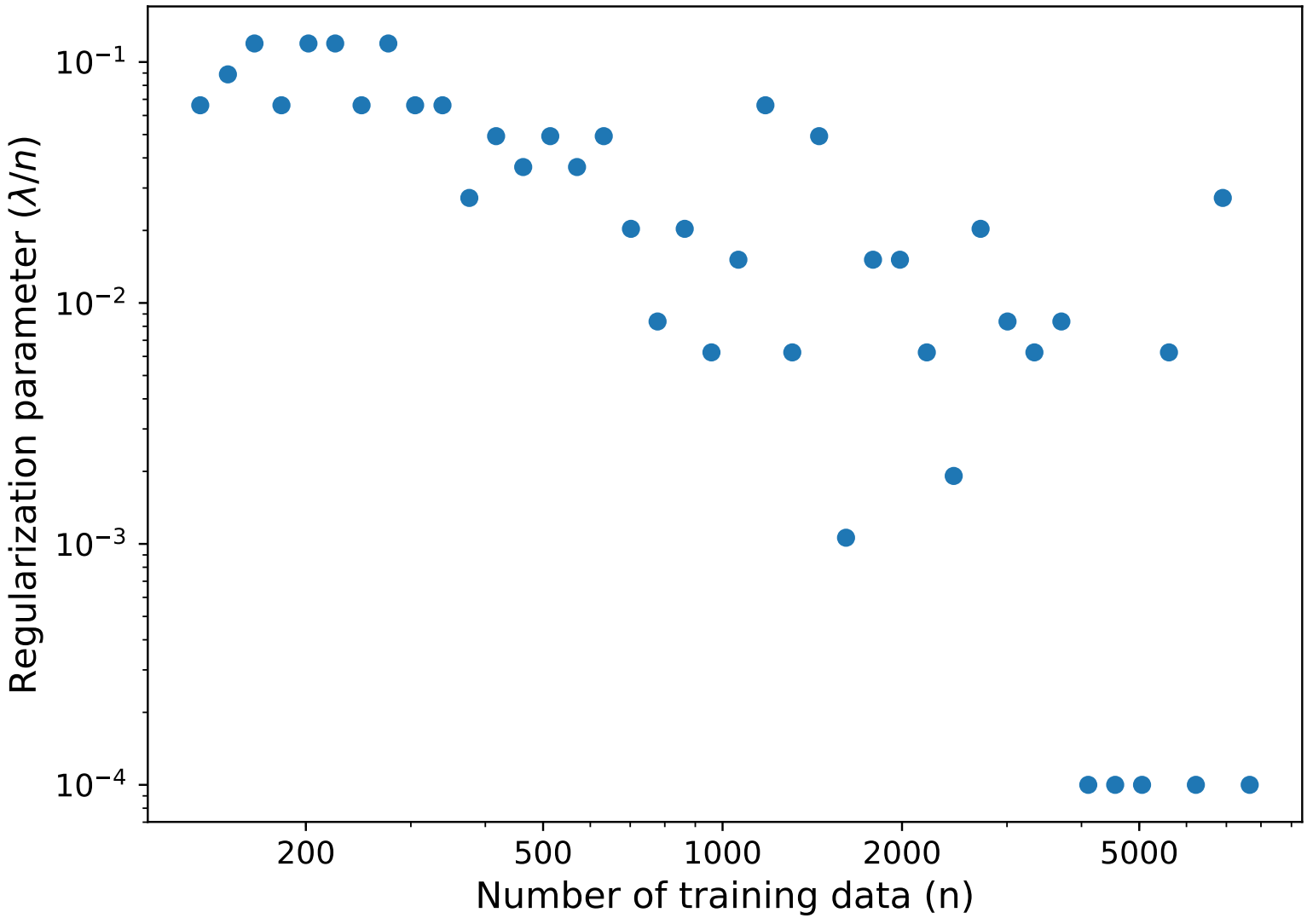
Temperature prediction via linear regression ($n = 202$)



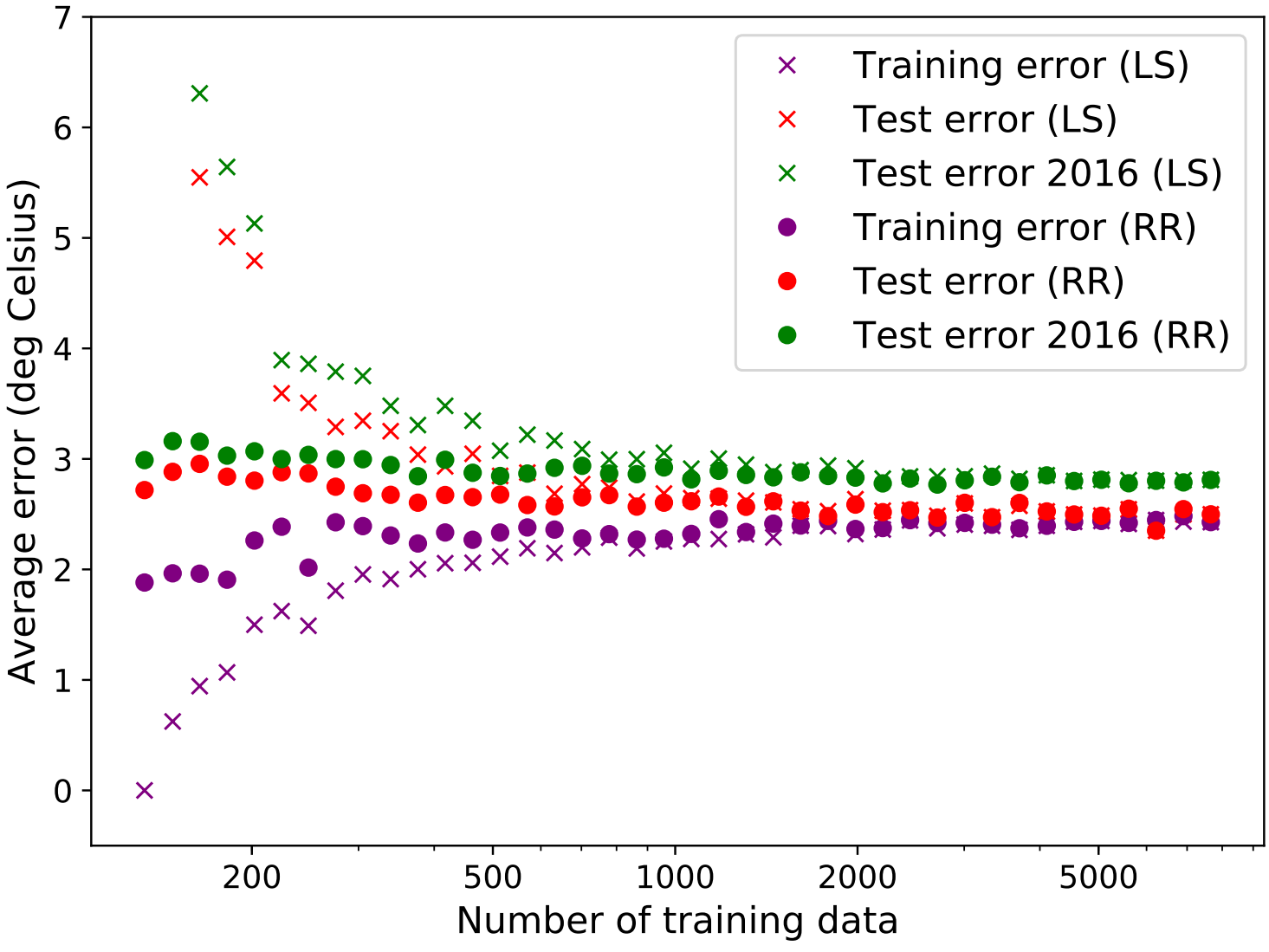
Temperature prediction via linear regression



Temperature prediction via linear regression



Temperature prediction via linear regression



Ridge-regression estimator

If $\vec{y}_{\text{train}} := X_{\text{train}}\vec{\beta}_{\text{true}} + \vec{z}_{\text{train}}$

$$\vec{\beta}_{\text{RR}} = V \begin{bmatrix} \frac{s_1^2}{s_1^2+\lambda} & 0 & \cdots & 0 \\ 0 & \frac{s_2^2}{s_2^2+\lambda} & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & \frac{s_p^2}{s_p^2+\lambda} \end{bmatrix} V^T \vec{\beta}_{\text{true}} + V \begin{bmatrix} \frac{s_1}{s_1^2+\lambda} & 0 & \cdots & 0 \\ 0 & \frac{s_2}{s_2^2+\lambda} & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & \frac{s_p}{s_p^2+\lambda} \end{bmatrix} U^T \vec{z}_{\text{train}}$$

where USV^T is the SVD of X_{train} and $s_1, \ldots, s_p$ are the singular values

Proof

$$\vec{\beta}_{\mathsf{RR}} = \left(X_{\mathsf{train}}^{T} X_{\mathsf{train}} + \lambda I\right)^{-1} X_{\mathsf{train}}^{T} \left(X_{\mathsf{train}} \vec{\beta}_{\mathsf{true}} + \vec{z}_{\mathsf{train}}\right)$$

Proof

$$\begin{aligned}\vec{\beta}_{\mathsf{RR}} &= \left(X_{\mathsf{train}}^T X_{\mathsf{train}} + \lambda I\right)^{-1} X_{\mathsf{train}}^T \left(X_{\mathsf{train}} \vec{\beta}_{\mathsf{true}} + \vec{z}_{\mathsf{train}}\right) \\ &= \left(V S^2 V^T + \lambda V V^T\right)^{-1} \left(V S^2 V^T \vec{\beta}_{\mathsf{true}} + V S U^T \vec{z}_{\mathsf{train}}\right)\end{aligned}$$

Proof

$$
\begin{aligned}
\vec{\beta}_{\mathsf{RR}} &= \left(X_{\mathsf{train}}^T X_{\mathsf{train}} + \lambda I\right)^{-1} X_{\mathsf{train}}^T \left(X_{\mathsf{train}} \vec{\beta}_{\mathsf{true}} + \vec{z}_{\mathsf{train}}\right) \\
&= \left(V S^2 V^T + \lambda V V^T\right)^{-1} \left(V S^2 V^T \vec{\beta}_{\mathsf{true}} + V S U^T \vec{z}_{\mathsf{train}}\right) \\
&= V \left(S^2 + \lambda I\right)^{-1} V^T \left(V S^2 V^T \vec{\beta}_{\mathsf{true}} + V S U^T \vec{z}_{\mathsf{train}}\right)
\end{aligned}
$$

Proof

$$
\begin{aligned}
\vec{\beta}_{\mathsf{RR}} &= \left(X_{\mathsf{train}}^T X_{\mathsf{train}} + \lambda I\right)^{-1} X_{\mathsf{train}}^T \left(X_{\mathsf{train}} \vec{\beta}_{\mathsf{true}} + \vec{z}_{\mathsf{train}}\right) \\
&= \left(V S^2 V^T + \lambda V V^T\right)^{-1} \left(V S^2 V^T \vec{\beta}_{\mathsf{true}} + V S U^T \vec{z}_{\mathsf{train}}\right) \\
&= V \left(S^2 + \lambda I\right)^{-1} V^T \left(V S^2 V^T \vec{\beta}_{\mathsf{true}} + V S U^T \vec{z}_{\mathsf{train}}\right) \\
&= V \left(S^2 + \lambda I\right)^{-1} S^2 V^T \vec{\beta}_{\mathsf{true}} + V \left(S^2 + \lambda I\right)^{-1} S U^T \vec{z}_{\mathsf{train}}
\end{aligned}
$$

Coefficient and test error

$$\vec{\beta}_{\mathsf{RR}} - \vec{\beta}_{\mathsf{true}} = -\lambda V \left(S^2 + \lambda I\right)^{-1} V^T \vec{\beta}_{\mathsf{true}} + V \left(S^2 + \lambda I\right)^{-1} S U^T \vec{z}_{\mathsf{train}}$$

Coefficient and test error

$$\vec{\beta}_{\mathrm{RR}} - \vec{\beta}_{\mathrm{true}} = -\lambda V \left(S^2 + \lambda I\right)^{-1} V^T \vec{\beta}_{\mathrm{true}} + V \left(S^2 + \lambda I\right)^{-1} S U^T \vec{z}_{\mathrm{train}}$$

$$\begin{aligned} y_{\mathrm{test}} - y_{\mathrm{RR}} &= \langle \vec{x}_{\mathrm{test}}, \vec{\beta}_{\mathrm{true}} - \vec{\beta}_{\mathrm{RR}} \rangle + z_{\mathrm{test}} \\ &= z_{\mathrm{test}} + \sum_{i=1}^{p} \frac{\langle \vec{x}_{\mathrm{test}}, \vec{v}_i \rangle \left(\lambda \left\langle \vec{\beta}_{\mathrm{true}}, \vec{v}_i \right\rangle - s_i \langle \vec{u}_i, \vec{z}_{\mathrm{train}} \rangle \right)}{s_i^2 + \lambda} \end{aligned}$$

Coefficient and test error

$$\vec{\beta}_{\mathrm{RR}} - \vec{\beta}_{\mathrm{true}} = -\lambda V \left(S^2 + \lambda I\right)^{-1} V^T \vec{\beta}_{\mathrm{true}} + V \left(S^2 + \lambda I\right)^{-1} S U^T \vec{z}_{\mathrm{train}}$$

$$\begin{aligned} y_{\mathrm{test}} - y_{\mathrm{RR}} &= \langle \vec{x}_{\mathrm{test}}, \vec{\beta}_{\mathrm{true}} - \vec{\beta}_{\mathrm{RR}} \rangle + z_{\mathrm{test}} \\ &= z_{\mathrm{test}} + \sum_{i=1}^{p} \frac{\langle \vec{x}_{\mathrm{test}}, \vec{v}_i \rangle \left(\lambda \left\langle \vec{\beta}_{\mathrm{true}}, \vec{v}_i \right\rangle - s_i \langle \vec{u}_i, \vec{z}_{\mathrm{train}} \rangle \right)}{s_i^2 + \lambda} \end{aligned}$$

$$y_{\mathrm{test}} - y_{\mathrm{LS}} = z_{\mathrm{test}} - \sum_{i=1}^{p} \frac{\langle \vec{x}_{\mathrm{test}}, \vec{v}_i \rangle \langle \vec{u}_i, \vec{z}_{\mathrm{train}} \rangle}{s_i}$$

Motivating applications

The singular-value decomposition

Principal component analysis

Dimensionality reduction

Linear regression

Ridge regression

Low-rank matrix estimation

Singular value decomposition

Every rank r real matrix $A \in R^{m \times n}$, has a singular-value decomposition (SVD) of the form

$$A = \begin{bmatrix} \vec{u}_1 & \vec{u}_2 & \cdots & \vec{u}_r \end{bmatrix} \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \cdots & s_r \end{bmatrix} \begin{bmatrix} \vec{v}_1^T \\ \vec{v}_2^T \\ \vdots \\ \vec{v}_r^T \end{bmatrix}$$

$$= USV^T$$

Connection between rank and SVD

Qualitative connection: rank = number of nonzero singular values

Quantitative connection: decomposition in rank-1 matrices

$$A = USV^T = \sum_{i=1}^{n} s_i \vec{u}_i \vec{v}_i^T$$

Vector space of matrices

$\mathbb{R}^{m\times n}$ matrices are a vector space

Inner product:

$$\langle A, B\rangle := \mathrm{tr}\left(A^T B\right), \quad A, B \in \mathbb{R}^{m\times n},$$

Norm:

$$||A||_{\mathrm{F}} := \sqrt{\mathrm{tr}\left(A^T A\right)} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2},$$

Orthonormal set

The rank-1 matrices $\vec{u}_i \vec{v}_i^T$ are orthonormal vectors

$$\left|\left|\vec{u}_i \vec{v}_i^T\right|\right|_F^2 = \text{tr}\left(\vec{v}_i \vec{u}_i^T \vec{u}_i \vec{v}_i^T\right)$$

$$\left\langle \vec{u}_i \vec{v}_i^T, \vec{u}_i \vec{v}_i^T \right\rangle = \text{tr}\left(\vec{v}_i \vec{u}_i^T \vec{u}_j \vec{v}_j^T\right)$$

Orthonormal set

The rank-1 matrices $\vec{u}_i \vec{v}_i^T$ are orthonormal vectors

$$\left|\left|\vec{u}_i \vec{v}_i^T\right|\right|_F^2 = \text{tr}\left(\vec{v}_i \vec{u}_i^T \vec{u}_i \vec{v}_i^T\right)$$
$$= \vec{v}_i^T \vec{v}_i \vec{u}_i^T \vec{u}_i = 1$$

$$\left\langle \vec{u}_i \vec{v}_i^T, \vec{u}_i \vec{v}_i^T \right\rangle = \text{tr}\left(\vec{v}_i \vec{u}_i^T \vec{u}_j \vec{v}_j^T\right)$$

Orthonormal set

The rank-1 matrices $\vec{u}_i \vec{v}_i^T$ are orthonormal vectors

$$\begin{aligned} \left|\left| \vec{u}_i \vec{v}_i^T \right|\right|_F^2 &= \text{tr}\left(\vec{v}_i \vec{u}_i^T \, \vec{u}_i \vec{v}_i^T \right) \\ &= \vec{v}_i^T \, \vec{v}_i \vec{u}_i^T \, \vec{u}_i = 1 \end{aligned}$$

$$\begin{aligned} \left\langle \vec{u}_i \vec{v}_i^T , \vec{u}_i \vec{v}_i^T \right\rangle &= \text{tr}\left(\vec{v}_i \vec{u}_i^T \, \vec{u}_j \vec{v}_j^T \right) \\ &= \vec{v}_i^T \, \vec{v}_j \vec{u}_j^T \, \vec{u}_i = 0, \text{ if } i \neq j \end{aligned}$$

Decomposition of matrix norm

$$
\begin{aligned}
||A||_{\mathrm{F}}^2 &= \sum_{i=1}^{n} \left|\left| s_i \vec{u}_i \vec{v}_i^{\,T} \right|\right|_{\mathrm{F}}^2, \quad \text{by Pythagoras's Theorem,} \\
&= \sum_{i=1}^{n} s_i^2, \quad \text{by homogeneity of norms.}
\end{aligned}
$$

Connection between rank and SVD

SVD decomposes a matrix into n components

Norm of each component equals corresponding singular value

A matrix is approximately rank r if last $n - r$ singular values are small with respect to first r

Connection between rank and SVD

SVD decomposes a matrix into n components

Norm of each component equals corresponding singular value

A matrix is approximately rank r if last $n - r$ singular values are small with respect to first r

Error of approximation given by last $n - r$ singular values

$$\left\| A - \sum_{i=1}^{r} s_i \vec{u}_i \vec{v}_i^T \right\|_F^2 = \left\| \sum_{i=r+1}^{n} s_i \vec{u}_i \vec{v}_i^T \right\|_F^2$$

$$= \sum_{i=r+1}^{n} s_i^2$$

Rank-r bilinear model

Certain people like certain movies: r factors

$$y[i,j] \approx \sum_{l=1}^{r} a_l[i] b_l[j]$$

For each factor l

- $a_l[i]$: movie i is positively (> 0), negatively (< 0) or not (≈ 0) associated to factor l
- $b_l[j]$: user j is positively (> 0), negatively (< 0) or not (≈ 0) associated to factor l

Rank-r bilinear model

In matrix form,

$$\begin{aligned} Y &\approx \begin{bmatrix} a_1 & a_2 & \cdots & a_r \end{bmatrix} \begin{bmatrix} b_1 & b_2 & \cdots & b_r \end{bmatrix}^T \\ &= AB^T \end{aligned}$$

To fit the model, we would like to solve

$$\min_{A \in \mathbb{R}^{m \times r}, B \in \mathbb{R}^{r \times n}} ||Y - AB||_F \qquad \text{subject to} \quad ||A_{:,1}||_2 = 1, \ldots, ||A_{:,r}||_2 = 1,$$

Best rank-r approximation

Let USV^T be the SVD of a matrix $A \in \mathbb{R}^{m \times n}$

The truncated SVD $U_{:,1:r} S_{1:r,1:r} V_{:,1:r}^T$ is the best rank-r approximation

$$U_{:,1:r} S_{1:r,1:r} V_{:,1:r}^T = \underset{\{\tilde{A} \mid \operatorname{rank}(\tilde{A}) = r\}}{\arg\min} \left|\left| A - \tilde{A} \right|\right|_F$$

Proof

Let $\widetilde{A}$ be an arbitrary matrix in $\mathbb{R}^{m \times n}$ with $\text{rank}(\widetilde{A}) = r$

Let $\widetilde{U} \in \mathbb{R}^{m \times k}$ be a matrix with orthonormal columns such that $\text{col}(\widetilde{U}) = \text{col}(\widetilde{A})$

$$\left|\left| U_{:,1:r} U_{:,1:r}^T A \right|\right|_F^2 = \sum_{i=1}^{n} \left|\left| \mathcal{P}_{\text{col}(U_{:,1:r})} A_{:,i} \right|\right|_2^2$$

Proof

Let $\widetilde{A}$ be an arbitrary matrix in $\mathbb{R}^{m \times n}$ with $\text{rank}(\widetilde{A}) = r$

Let $\widetilde{U} \in \mathbb{R}^{m \times k}$ be a matrix with orthonormal columns such that $\text{col}(\widetilde{U}) = \text{col}(\widetilde{A})$

$$\left|\left| U_{:,1:r} U_{:,1:r}^T A \right|\right|_F^2 = \sum_{i=1}^{n} \left|\left| \mathcal{P}_{\text{col}(U_{:,1:r})} A_{:,i} \right|\right|_2^2$$

$$\geq \sum_{i=1}^{n} \left|\left| \mathcal{P}_{\text{col}(\widetilde{U})} A_{:,i} \right|\right|_2^2$$

Proof

Let $\widetilde{A}$ be an arbitrary matrix in $\mathbb{R}^{m\times n}$ with $\text{rank}(\widetilde{A}) = r$

Let $\widetilde{U} \in \mathbb{R}^{m\times k}$ be a matrix with orthonormal columns such that $\text{col}(\widetilde{U}) = \text{col}(\widetilde{A})$

$$\begin{aligned}
\left|\left|U_{:,1:r}U_{:,1:r}^T A\right|\right|_F^2 &= \sum_{i=1}^{n} \left|\left|\mathcal{P}_{\text{col}(U_{:,1:r})} A_{:,i}\right|\right|_2^2 \\
&\geq \sum_{i=1}^{n} \left|\left|\mathcal{P}_{\text{col}(\widetilde{U})} A_{:,i}\right|\right|_2^2 \\
&= \left|\left|\widetilde{U}\widetilde{U}^T A\right|\right|_F^2
\end{aligned}$$

Optimal subspace for orthogonal projection

Given a set of vectors $\vec{a}_1, \vec{a}_2, \ldots \vec{a}_n$ and a fixed dimension $k \leq n$, the SVD of

$$A := \begin{bmatrix} \vec{a}_1 & \vec{a}_2 & \cdots & \vec{a}_n \end{bmatrix} \in \mathbb{R}^{m \times n}$$

provides the k-dimensional subspace that captures the most energy

$$\sum_{i=1}^{n} \left|\left| \mathcal{P}_{\text{span}(\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k)} \, \vec{a}_i \right|\right|_2^2 \geq \sum_{i=1}^{n} ||\mathcal{P}_{\mathcal{S}} \, \vec{a}_i||_2^2$$

for any subspace $\mathcal{S}$ of dimension k

Orthogonal column spaces

If the column spaces of $A, B \in \mathbb{R}^{m \times n}$ are orthogonal then

$$||A + B||_{\mathsf{F}}^2 = ||A||_{\mathsf{F}}^2 + ||B||_{\mathsf{F}}^2$$

Corollary:

$$||A||_{\mathsf{F}}^2 = ||A + B||_{\mathsf{F}}^2 - ||B||_{\mathsf{F}}^2$$

Proof

$\text{col}\left(A - \widetilde{U}\widetilde{U}^T A\right)$ is orthogonal to $\text{col}\left(\widetilde{A}\right) = \text{col}\left(\widetilde{U}\right)$

$$\left|\left|A - \widetilde{A}\right|\right|_{\text{F}}^2 = \left|\left|A - \widetilde{U}\widetilde{U}^T A\right|\right|_{\text{F}}^2 + \left|\left|\widetilde{A} - \widetilde{U}\widetilde{U}^T A\right|\right|_{\text{F}}^2$$

Proof

$\text{col}\left(A - \widetilde{U}\widetilde{U}^T A\right)$ is orthogonal to $\text{col}\left(\widetilde{A}\right) = \text{col}\left(\widetilde{U}\right)$

$$\begin{aligned}\left|\left|A - \widetilde{A}\right|\right|_F^2 &= \left|\left|A - \widetilde{U}\widetilde{U}^T A\right|\right|_F^2 + \left|\left|\widetilde{A} - \widetilde{U}\widetilde{U}^T A\right|\right|_F^2 \\ &\geq \left|\left|A - \widetilde{U}\widetilde{U}^T A\right|\right|_F^2\end{aligned}$$

Proof

$\text{col}\left(A - \widetilde{U}\widetilde{U}^T A\right)$ is orthogonal to $\text{col}\left(\widetilde{A}\right) = \text{col}\left(\widetilde{U}\right)$

$$\begin{aligned}
\left|\left|A - \widetilde{A}\right|\right|_F^2 &= \left|\left|A - \widetilde{U}\widetilde{U}^T A\right|\right|_F^2 + \left|\left|\widetilde{A} - \widetilde{U}\widetilde{U}^T A\right|\right|_F^2 \\
&\geq \left|\left|A - \widetilde{U}\widetilde{U}^T A\right|\right|_F^2 \\
&= ||A||_F^2 - \left|\left|\widetilde{U}\widetilde{U}^T A\right|\right|_F^2
\end{aligned}$$

Proof

$\text{col}\left(A - \widetilde{U}\widetilde{U}^T A\right)$ is orthogonal to $\text{col}\left(\widetilde{A}\right) = \text{col}\left(\widetilde{U}\right)$

$$\begin{aligned}
\left|\left|A - \widetilde{A}\right|\right|_F^2 &= \left|\left|A - \widetilde{U}\widetilde{U}^T A\right|\right|_F^2 + \left|\left|\widetilde{A} - \widetilde{U}\widetilde{U}^T A\right|\right|_F^2 \\
&\geq \left|\left|A - \widetilde{U}\widetilde{U}^T A\right|\right|_F^2 \\
&= ||A||_F^2 - \left|\left|\widetilde{U}\widetilde{U}^T A\right|\right|_F^2 \\
&\geq ||A||_F^2 - \left|\left|U_{:,1:r} U_{:,1:r}^T A\right|\right|_F^2
\end{aligned}$$

Proof

$\text{col}\left(A - \widetilde{U}\widetilde{U}^T A\right)$ is orthogonal to $\text{col}\left(\widetilde{A}\right) = \text{col}\left(\widetilde{U}\right)$

$$\begin{aligned}
\left|\left|A - \widetilde{A}\right|\right|_F^2 &= \left|\left|A - \widetilde{U}\widetilde{U}^T A\right|\right|_F^2 + \left|\left|\widetilde{A} - \widetilde{U}\widetilde{U}^T A\right|\right|_F^2 \\
&\geq \left|\left|A - \widetilde{U}\widetilde{U}^T A\right|\right|_F^2 \\
&= ||A||_F^2 - \left|\left|\widetilde{U}\widetilde{U}^T A\right|\right|_F^2 \\
&\geq ||A||_F^2 - \left|\left|U_{:,1:r}U_{:,1:r}^T A\right|\right|_F^2 \\
&= \left|\left|A - U_{:,1:r}U_{:,1:r}^T A\right|\right|_F^2
\end{aligned}$$

Collaborative filtering

- Movielens data
- Ratings between 1 and 10
- 100 users and movies with more ratings
- 6,031 out of 10^4 ratings are observed
- Test set with 10^3
- Validation set with $\max\{n_{\text{train}}, 400\}$

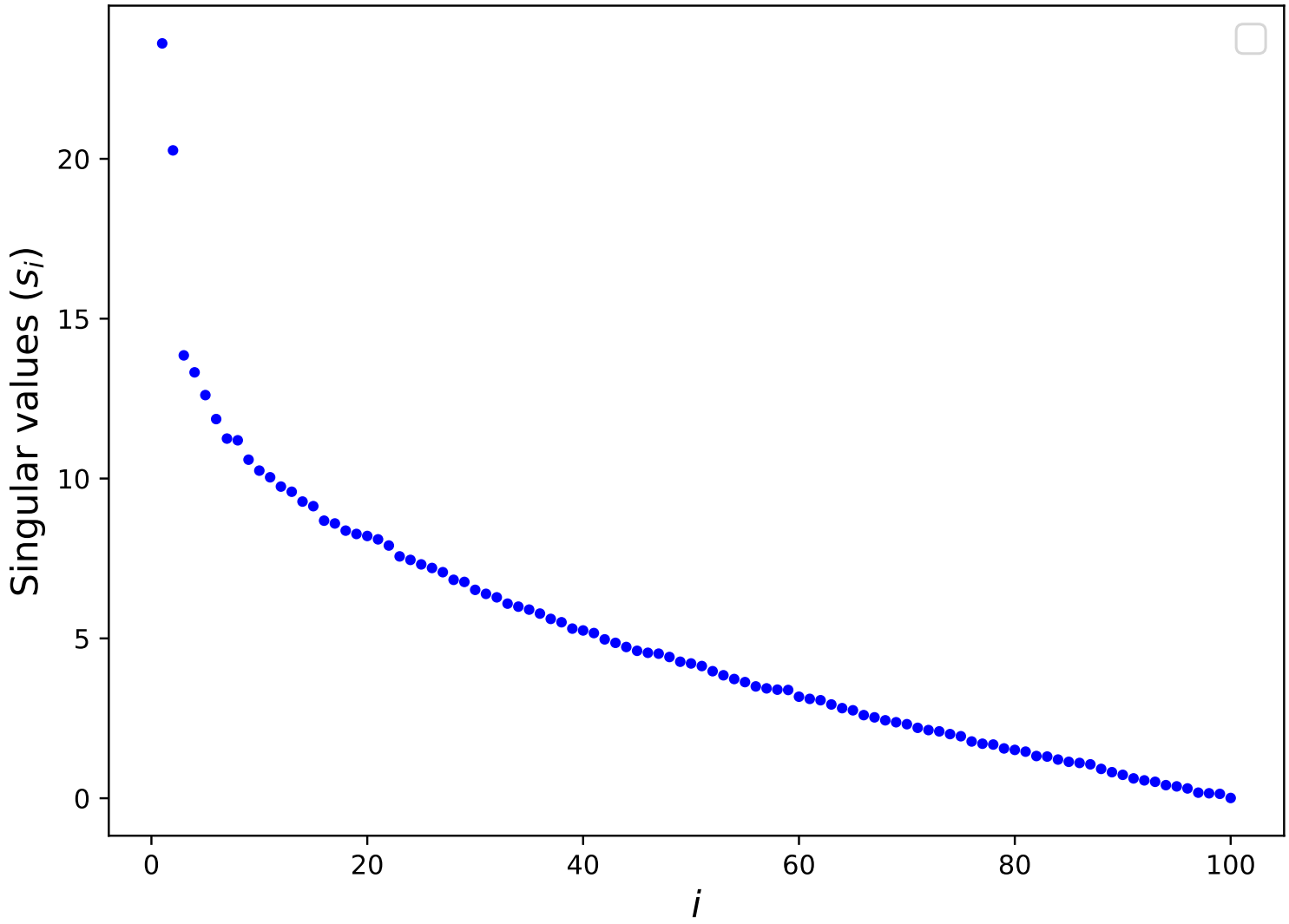
Bilinear model

Incorporates average rating μ

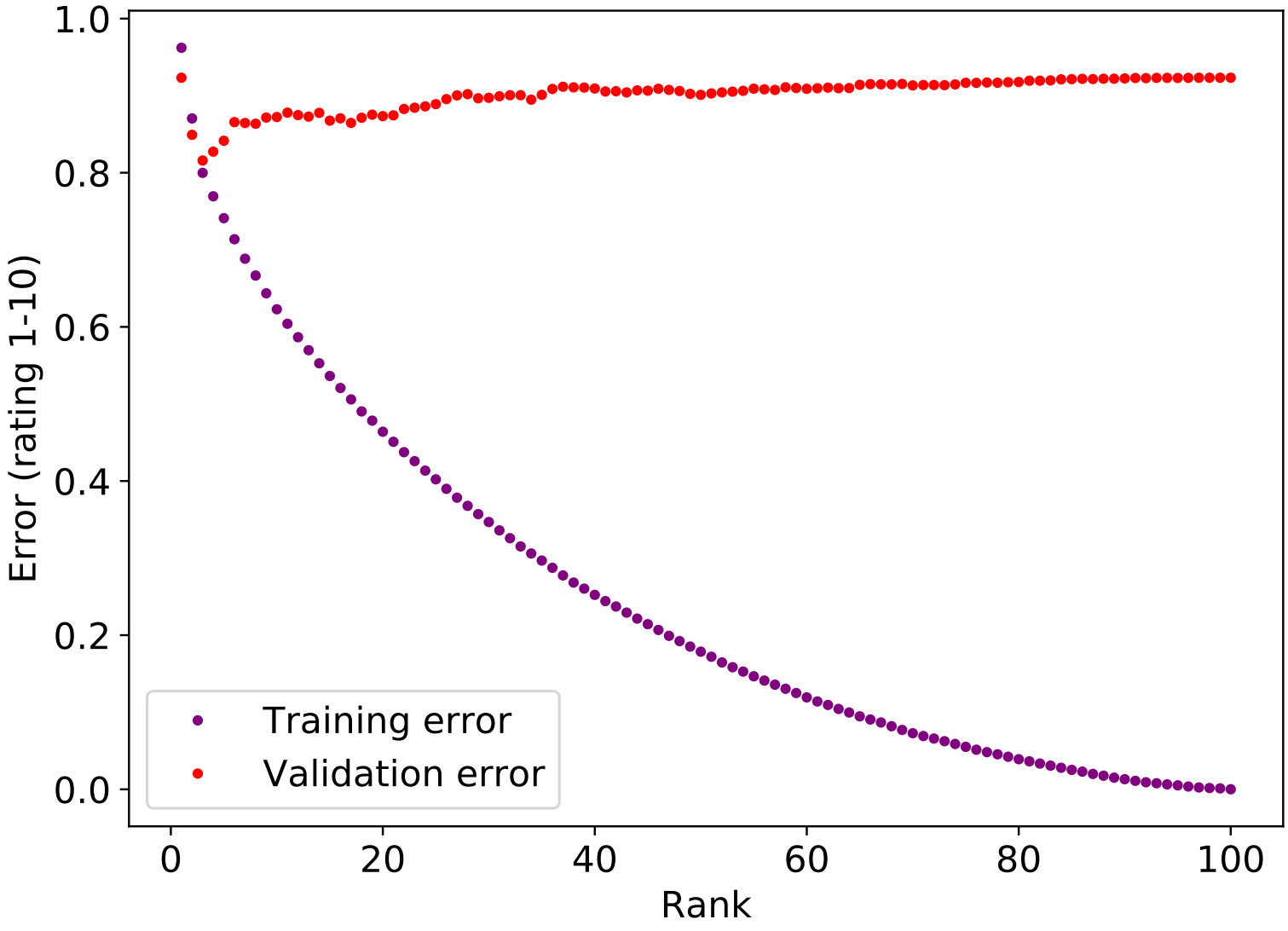
$$Y \approx AB + \mu$$

1. Set μ equal to average rating
2. Subtract average rating from all entries, set unobserved entries to 0
3. Compute SVD of centered matrix
4. Set $A := U_{:,1:r}$, $B := S_{1:r,1:r} V^T_{:,1:r}$

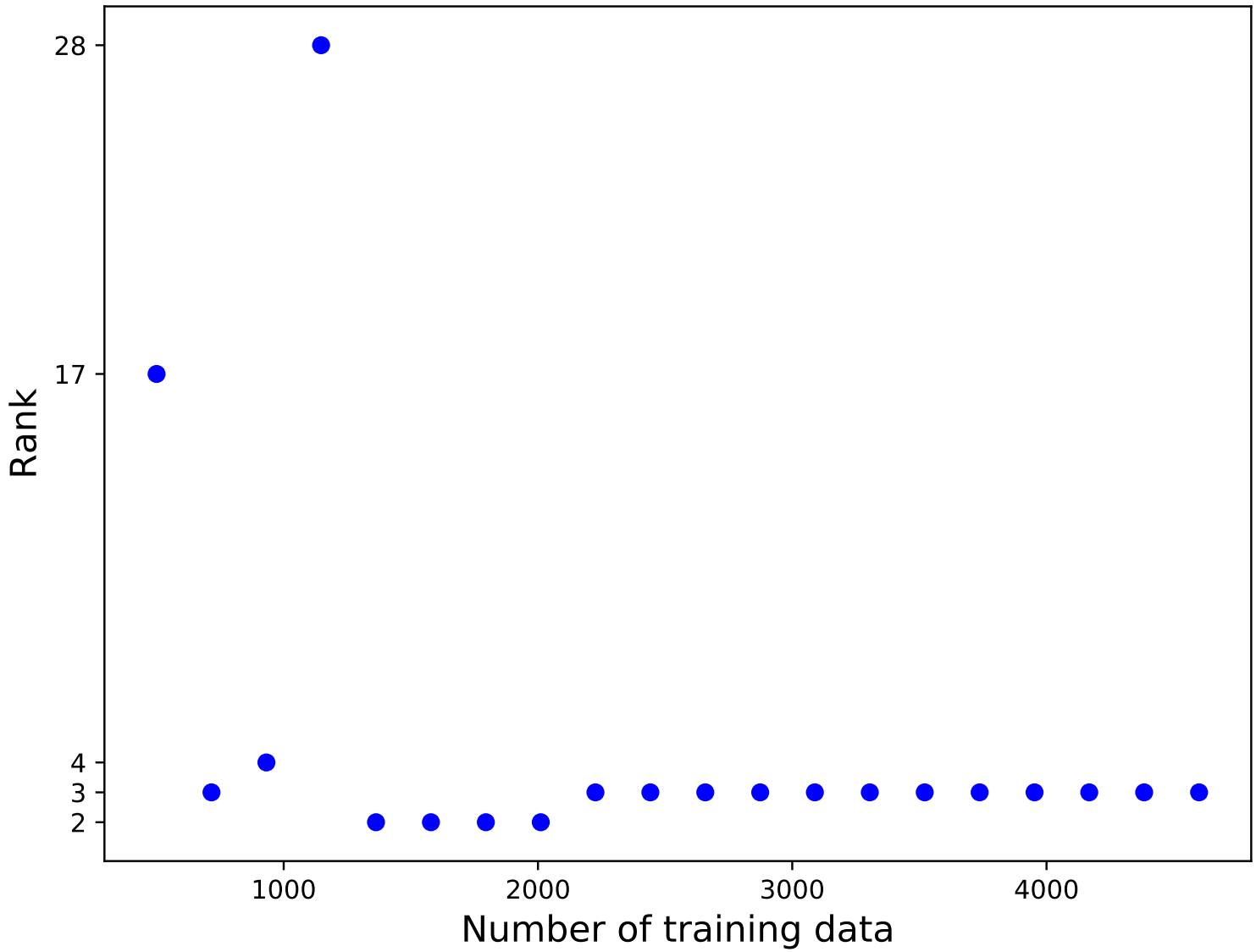
SVD of matrix ($n_{\text{train}} := 4,600$)



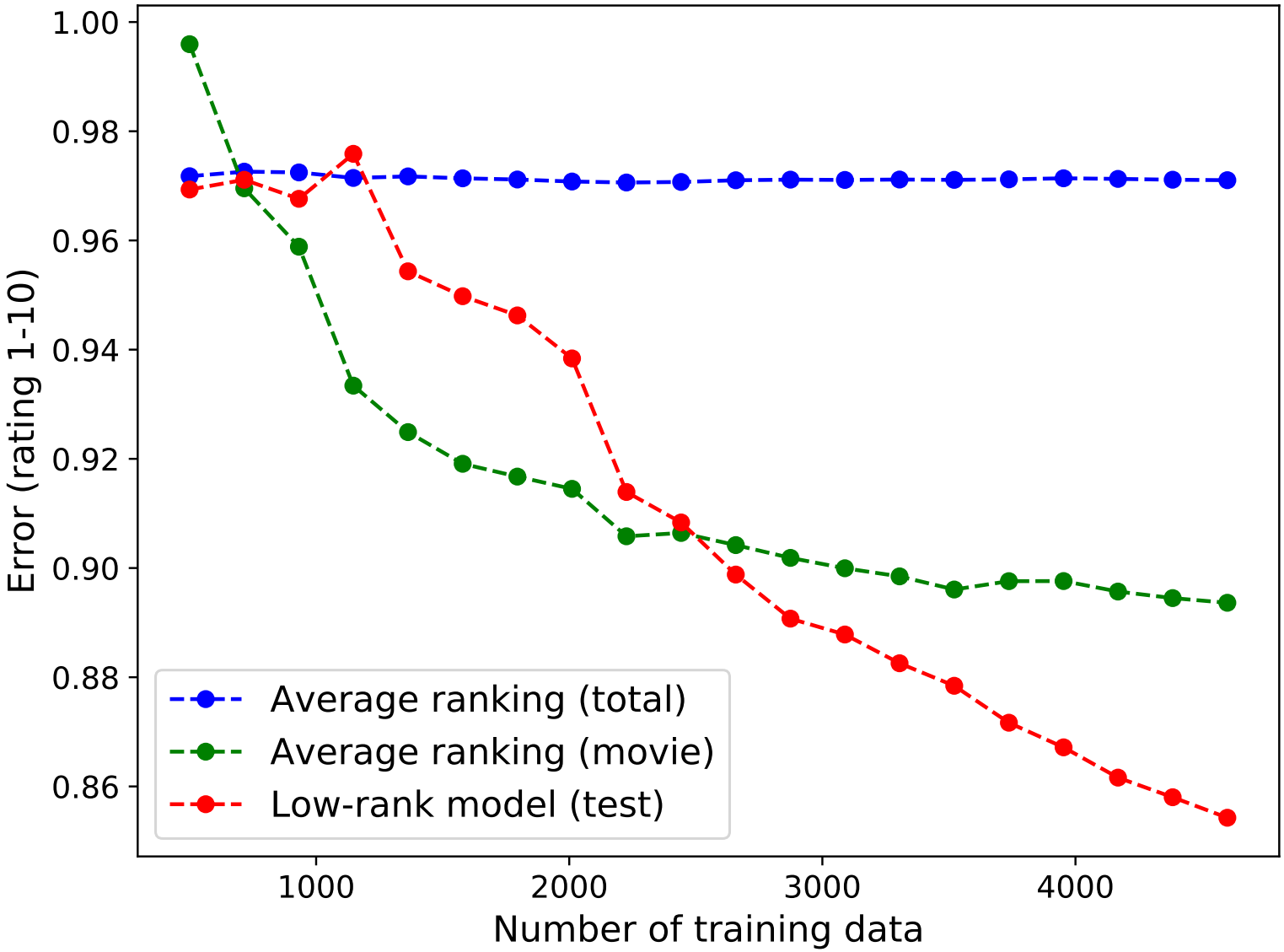
Results ($n_{\text{train}} := 4,600$)



Selected rank



Results


1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86
Error (rating 1-10)
1000
2000
3000
4000
Number of training data
Average ranking (total)
Average ranking (movie)
Low-rank model (test)

Rank-r bilinear model

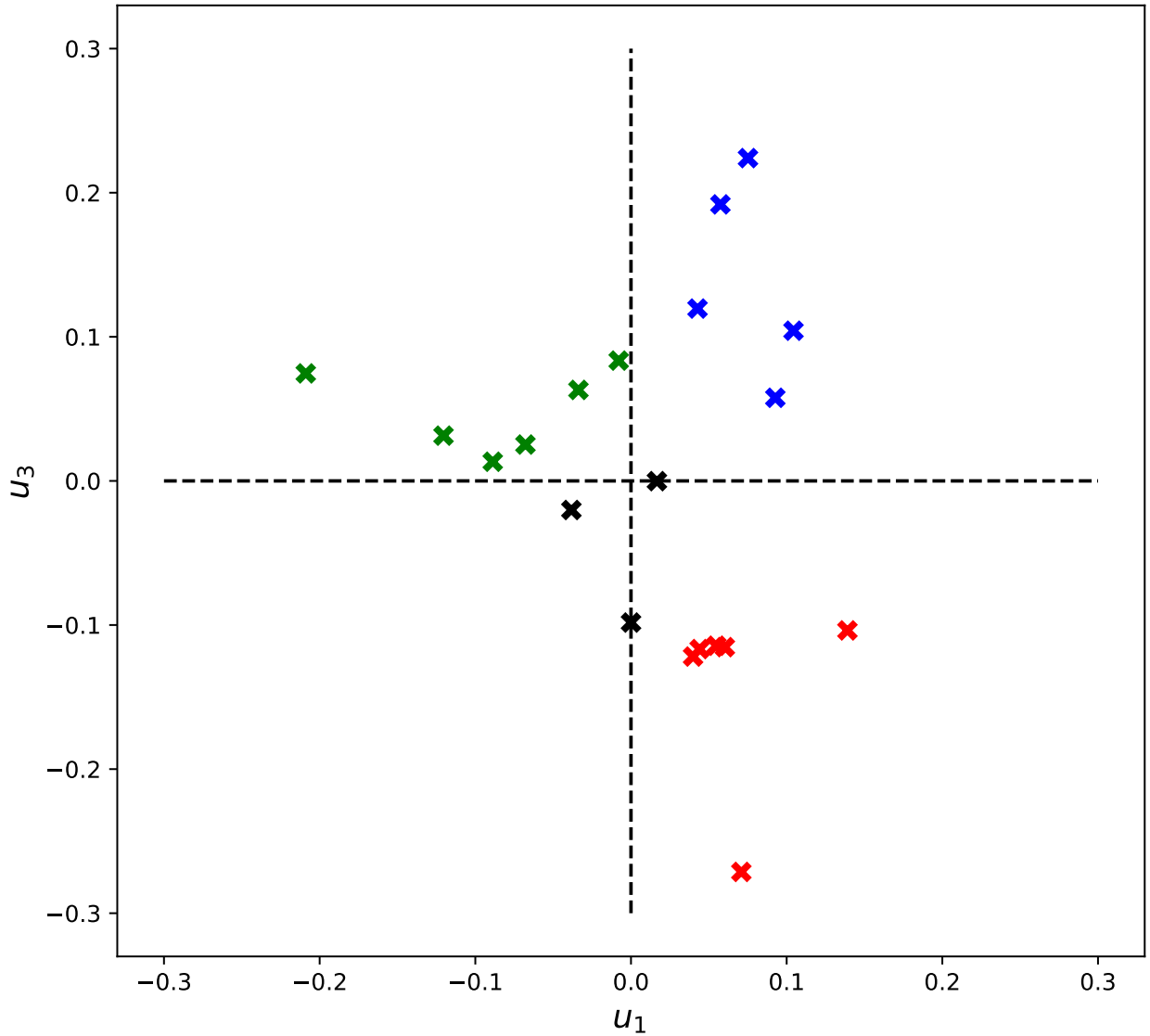
Certain people like certain movies: r factors

$$y[i,j] \approx \sum_{l=1}^{r} a_l[i] b_l[j]$$

For each factor l

- $a_l[i]$: movie i is positively (> 0), negatively (< 0) or not (≈ 0) associated to factor l
- $b_l[j]$: user j is positively (> 0), negatively (< 0) or not (≈ 0) associated to factor l

Rank-3 bilinear model



Rank-3 bilinear model

u_1	u_2	u_3	
-0.03	-0.15	0.06	Forrest Gump
0.14	-0.15	-0.10	Pulp Fiction
0.05	-0.15	-0.11	The Shawshank Redemption
0.04	-0.16	-0.12	Silence of the Lambs
0.09	-0.22	0.06	Star Wars Ep. IV
-0.12	-0.14	0.03	Jurassic Park
0.10	-0.14	0.10	The Matrix
-0.07	-0.10	0.03	Toy Story
0.04	-0.13	-0.12	Schindler's List
-0.01	-0.03	0.08	Terminator 2
0.08	-0.2	0.22	Star Wars Ep. V
0.00	-0.12	-0.10	Braveheart
0.02	-0.12	0.00	Back to the Future
0.07	-0.09	-0.27	Fargo
0.04	-0.23	0.12	Raiders of the Lost Ark
0.06	-0.13	-0.12	American Beauty
-0.21	0.07	0.08	Independence Day
0.06	-0.16	0.19	Star Wars Ep. VI
-0.09	-0.08	0.01	Aladdin
-0.04	-0.08	-0.02	The Fugitive