ORF307 – Optimization

5. Multi-objective least squares

Bartolomeo Stellato — Spring 2024

Ed Forum

- Would it be possible to post lecture notes a few weeks in advance, in case we ever want to review material ahead of class? —> *Last year's material is available at:* https://stellato.io/teaching/orf307/2023/

- Can you go over autoregressive time series models again?

Recap

Least squares data fitting

Vector form

Express problems with N-vectors

- $y^{\mathrm{d}} = (y^{(1)}, \dots, y^{(N)})$, vector of outcomes
- $\hat{y}^{\mathrm{d}} = (\hat{y}^{(1)}, \dots, \hat{y}^{(N)})$, vector of predictions
- $r^{\mathrm{d}} = (r^{(1)}, \dots, r^{(N)})$, vector of residuals

$\longrightarrow$

Goal

minimize $\quad \|r^{\mathrm{d}}\|^2$

We can write $\hat{y}^{(i)} = \hat{f}(x^{(i)})$ in terms of parameters θ_i

$$\hat{y}^{(i)} = A_{i1}\theta_1 + \cdots + A_{ip}\theta_p, \qquad A_{ij} = f_j(x^{(i)}) \qquad \longrightarrow \qquad \hat{y}^{\mathrm{d}} = A\theta$$

Least squares problem

minimize $\quad \|r^{\mathrm{d}}\|^2 = \|y^{\mathrm{d}} - \hat{y}^{\mathrm{d}}\|^2 = \|y^{\mathrm{d}} - A\theta\|^2 = \|A\theta - y^{\mathrm{d}}\|^2$

Solution

$$(A^T A)\theta^\star = A^T y^{\mathrm{d}}$$

Auto-regressive time series model

$z_1, z_2, \ldots$ is a **time series**

auto-regressive (AR) prediction model

$$\hat{z}_{t+1} = \theta_1 z_t + \cdots + \theta_M z_{t-M+1}, \quad t = M, M+1, \ldots$$

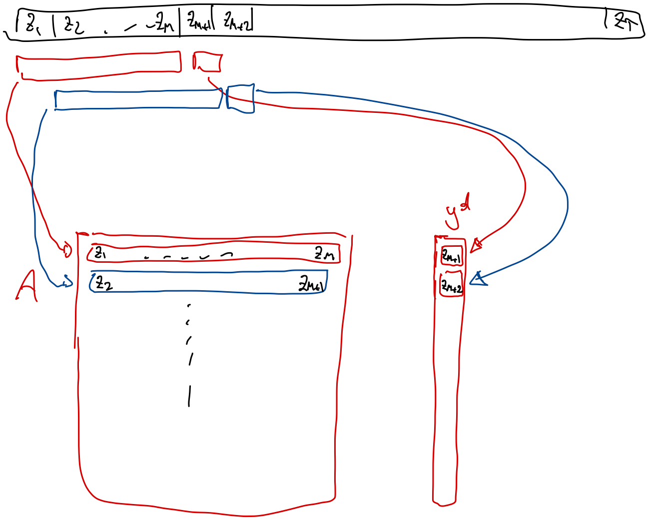
(predict $\hat{z}_{t+1}$ based on previous M values, where M is the memory)

Goal: Chose θ to minimize sum of squares of prediction errors

$$(\hat{z}_{M+1} - z_{M+1})^2 + \cdots + (\hat{z}_T - z_T)^2$$

General data fitting form

$$y^{(i)} = z_{M+i}, \qquad x^{(i)} = (z_{M+i-1}, \ldots, z_i), \quad i = 1, \ldots, T-M$$

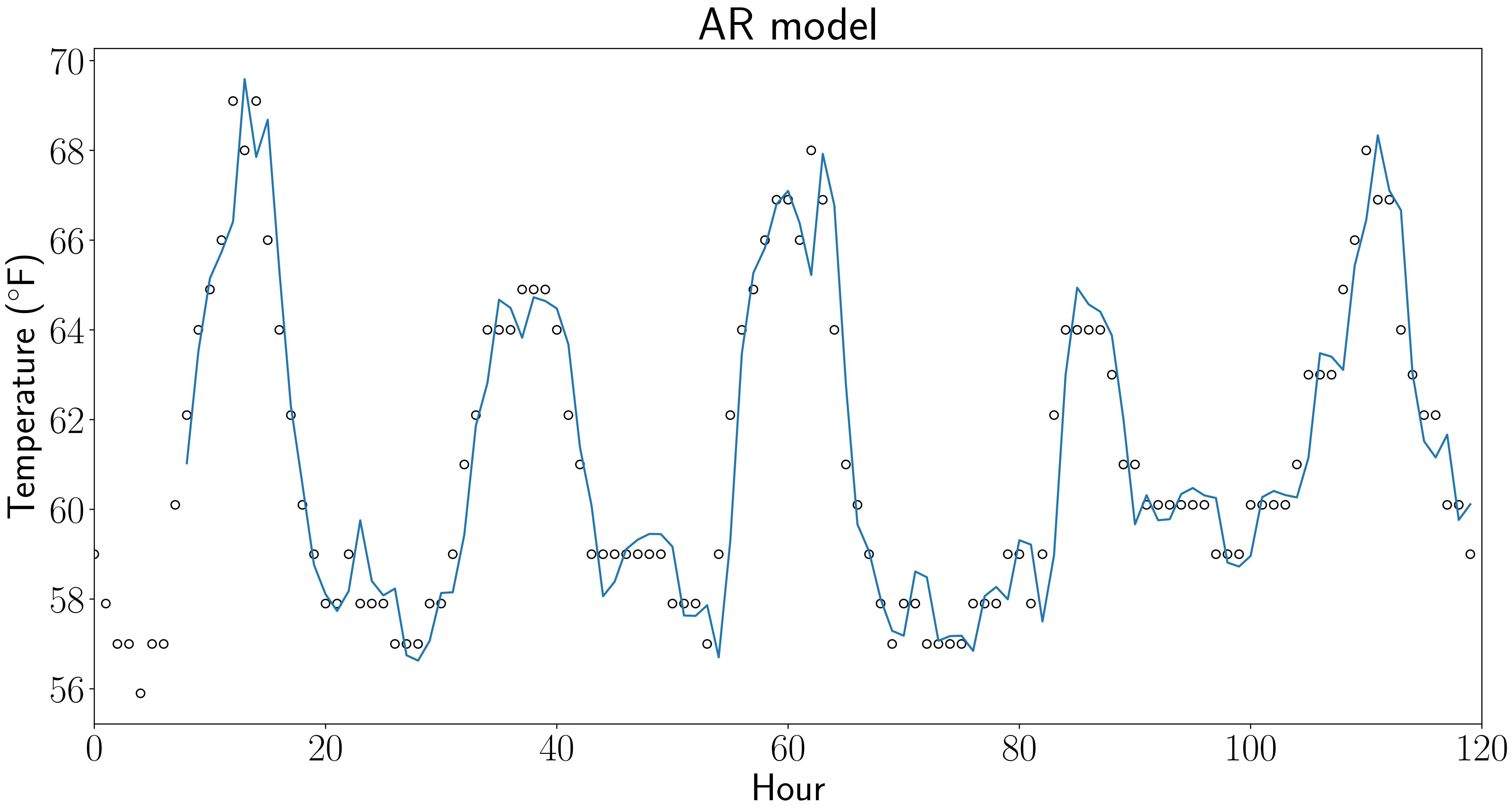


$$\longrightarrow \quad A^T A \theta^* = A^T y^d \rightarrow \boxed{\theta^*}$$

Auto-regressive time series model

5 days hourly temperature at Los Angeles International Airport (LAX)

- Previous hour: $\hat{z}_{t+1} = z_t$, MSE 1.35
- 24 hours before: $\hat{z}_{t+1} = z_{t-23}$, MSE = 3.00
- AR model with $M = 8$, MSE = 1.02



Today's lecture

Multi-objective least squares

- Multi-objective least squares problem
- Control
- Estimation
- Regularized data fitting

Multi-objective least squares problem

Multi-objective least squares

Goal choose n-vector x such that
k norm squared objectives are small

$$J_1 = \|A_1 x - b_1\|^2$$
$$\vdots$$
$$J_k = \|A_k x - b_k\|^2$$

A_i are $m_i \times n$ matrices and b_i are m_i-vectors for $i = 1, \ldots, k$

Multi-objective least squares

Goal choose n-vector x such that
k norm squared objectives are small

$$J_1 = \|A_1 x - b_1\|^2$$
$$\vdots$$
$$J_k = \|A_k x - b_k\|^2$$

A_i are $m_i \times n$ matrices and b_i are m_i-vectors for $i = 1, \ldots, k$

J_i are the objectives in a *multi-objective (-criterion) optimization problem*

Could choose x to minimize
any one J_i, but we want
to make them all small

Weighted sum objective

Choose positive weights $\lambda_1, \ldots, \lambda_k$ and form *weighted sum objective*

$$\begin{aligned} J &= \lambda_1 J_1 + \cdots + \lambda_k J_k \\ &= \lambda_1 \|A_1 x - b_1\|^2 + \cdots + \lambda_k \|A_k x - b_k\|^2 \end{aligned}$$

Choose x to minimize J

Weighted sum objective

Choose positive weights $\lambda_1, \ldots, \lambda_k$ and form *weighted sum objective*

$$J = \lambda_1 J_1 + \cdots + \lambda_k J_k$$
$$= \lambda_1 \|A_1 x - b_1\|^2 + \cdots + \lambda_k \|A_k x - b_k\|^2$$

Choose x to minimize J

Primary objective

- Often $\lambda_1 = 1$ and J_1 is the **primary objective**
- **Interpretation** λ_i is how much we care about J_i being small, relative to J_1

Weighted sum objective

Choose positive weights $\lambda_1, \ldots, \lambda_k$ and form *weighted sum objective*

$$J = \lambda_1 J_1 + \cdots + \lambda_k J_k$$
$$= \lambda_1 \|A_1 x - b_1\|^2 + \cdots + \lambda_k \|A_k x - b_k\|^2$$

Choose x to minimize J

Primary objective

- Often $\lambda_1 = 1$ and J_1 is the **primary objective**
- **Interpretation** λ_i is how much we care about J_i being small, relative to J_1

Bi-criterion optimization

$$J_1 + \lambda J_2 = \|A_1 x - b_1\|^2 + \lambda \|A_2 x - b_2\|^2$$

Weighted sum minimization as regular least squares

$r^1 = (r^1_1, r^1_2)$

$r^2 = (r^2_1, r^2_2)$

$(r^1_1)^2 + (r^1_2)^2 + (r^2_1)^2 + (r^2_2)^2$

$\|r^1\|^2 + \|r^2\|^2 = \left\| \begin{bmatrix} r^1 \\ r^2 \end{bmatrix} \right\|^2$

$$J = \lambda_1 \|\overbrace{A_1 x - b_1}^{r^1}\|^2 + \cdots + \lambda_k \|A_k x - b_k\|^2$$

$$= \left\| \begin{bmatrix} \sqrt{\lambda_1}(A_1 x - b_1) \\ \vdots \\ \sqrt{\lambda_k}(A_k x - b_k) \end{bmatrix} \right\|^2$$

stack objectives

Weighted sum minimization as regular least squares

$A_1 \in \mathbb{R}^{m_1 \times n}$

$A_2 \in \mathbb{R}^{m_2 \times n}$

$$J = \lambda_1 \|A_1 x - b_1\|^2 + \cdots + \lambda_k \|A_k x - b_k\|^2$$

$$= \left\| \begin{bmatrix} \sqrt{\lambda_1}(A_1 x - b_1) \\ \vdots \\ \sqrt{\lambda_k}(A_k x - b_k) \end{bmatrix} \right\|^2$$

stack objectives

Regular (single-criterion) least squares

$$J = \|\tilde{A}x - \tilde{b}\|^2$$

$\tilde{A}^T = [\sqrt{\lambda_1} A_1^T \ \sqrt{\lambda_2} A_2^T \ \ldots, \sqrt{\lambda_k} A_k^T]$

$\tilde{A} \in \mathbb{R}^{(m_1 + m_2 + \cdots + m_k) \times n}$

$$\tilde{A} = \begin{bmatrix} \sqrt{\lambda_1} A_1 \\ \vdots \\ \sqrt{\lambda_k} A_k \end{bmatrix}, \qquad \tilde{b} = \begin{bmatrix} \sqrt{\lambda_1} b_1 \\ \vdots \\ \sqrt{\lambda_k} b_k \end{bmatrix}$$

Weighted sum solution

Assuming the columns of $\tilde{A}$ are linearly independent

$$(\tilde{A}^T\tilde{A})x^\star = \tilde{A}^T\tilde{b}$$

$$(\lambda_1 A_1^T A_1 + \cdots + \lambda_k A_k^T A_k)x^\star = (\lambda_1 A_1^T b_1 + \cdots + \lambda_k A_k^T b_k)$$

Weighted sum solution

Assuming the columns of $\tilde{A}$ are linearly independent

$$(\tilde{A}^T\tilde{A})x^\star = \tilde{A}^T\tilde{b}$$

$$(\lambda_1 A_1^T A_1 + \cdots + \lambda_k A_k^T A_k)x^\star = (\lambda_1 A_1^T b_1 + \cdots + \lambda_k A_k^T b_k)$$

Remarks

- Can compute $x^\star$ via the Cholesky factorization of $\tilde{A}^T\tilde{A}$
- A_i can be wide or have dependent columns ($\tilde{A}$ can't)

Optimal trade-off curve

Bi-criterion problem

minimize $\quad J_1(x) + \lambda J_2(x) \quad \longrightarrow \quad x^\star(\lambda)$

Optimal trade-off curve

Bi-criterion problem

minimize $J_1(x) + \lambda J_2(x)$ $\longrightarrow$ $x^\star(\lambda)$

Pareto optimal $x^\star(\lambda)$

There is no point z that satisfies

$J_1(z) \leq J_1(x^\star(\lambda))$ and $J_2(z) \leq J_2(x^\star(\lambda))$

with one of the inequalities holding strictly
(no other point beats $x^\star$ on both objectives)

Optimal trade-off curve

Bi-criterion problem

minimize $J_1(x) + \lambda J_2(x) \quad \longrightarrow \quad x^\star(\lambda)$

Pareto optimal $x^\star(\lambda)$

There is no point z that satisfies

$$J_1(z) \leq J_1(x^\star(\lambda)) \quad \text{and} \quad J_2(z) \leq J_2(x^\star(\lambda))$$

with one of the inequalities holding strictly
(no other point beats $x^\star$ on both objectives)

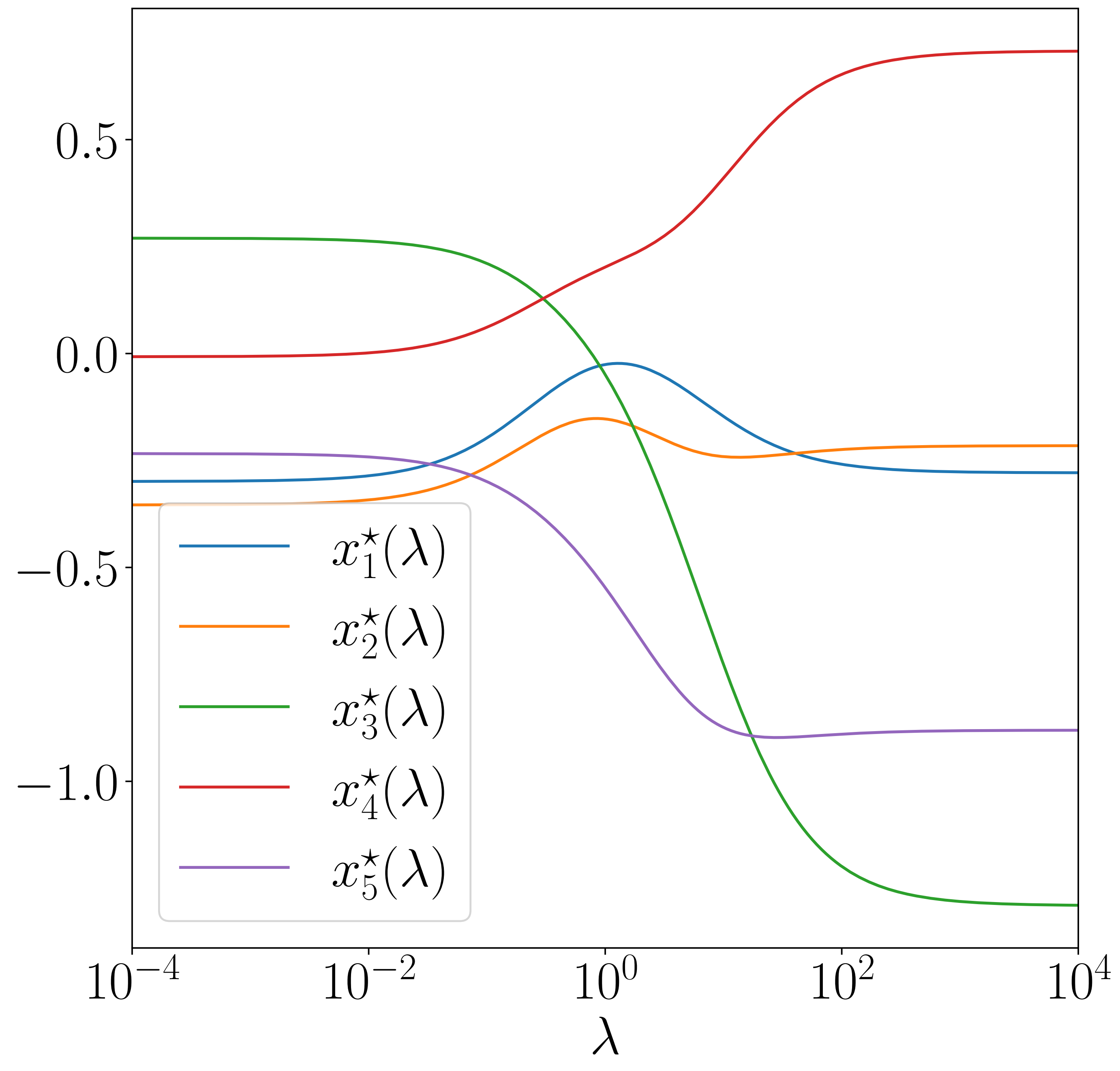
Optimal trade-off curve

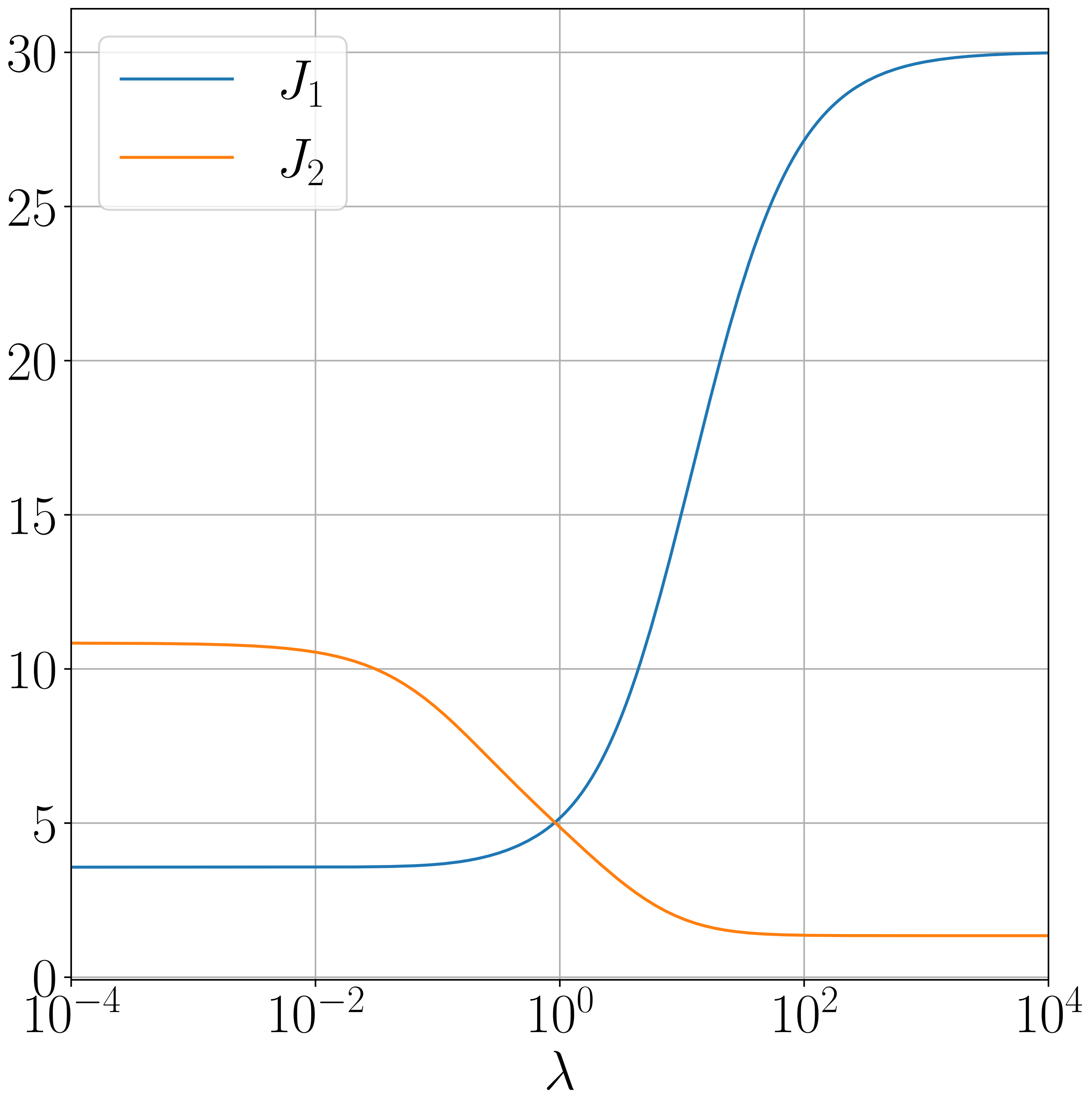
$$(J_1(x^\star(\lambda)), J_2(x^\star(\lambda)), \qquad \lambda > 0$$

Optimal trade-off curve

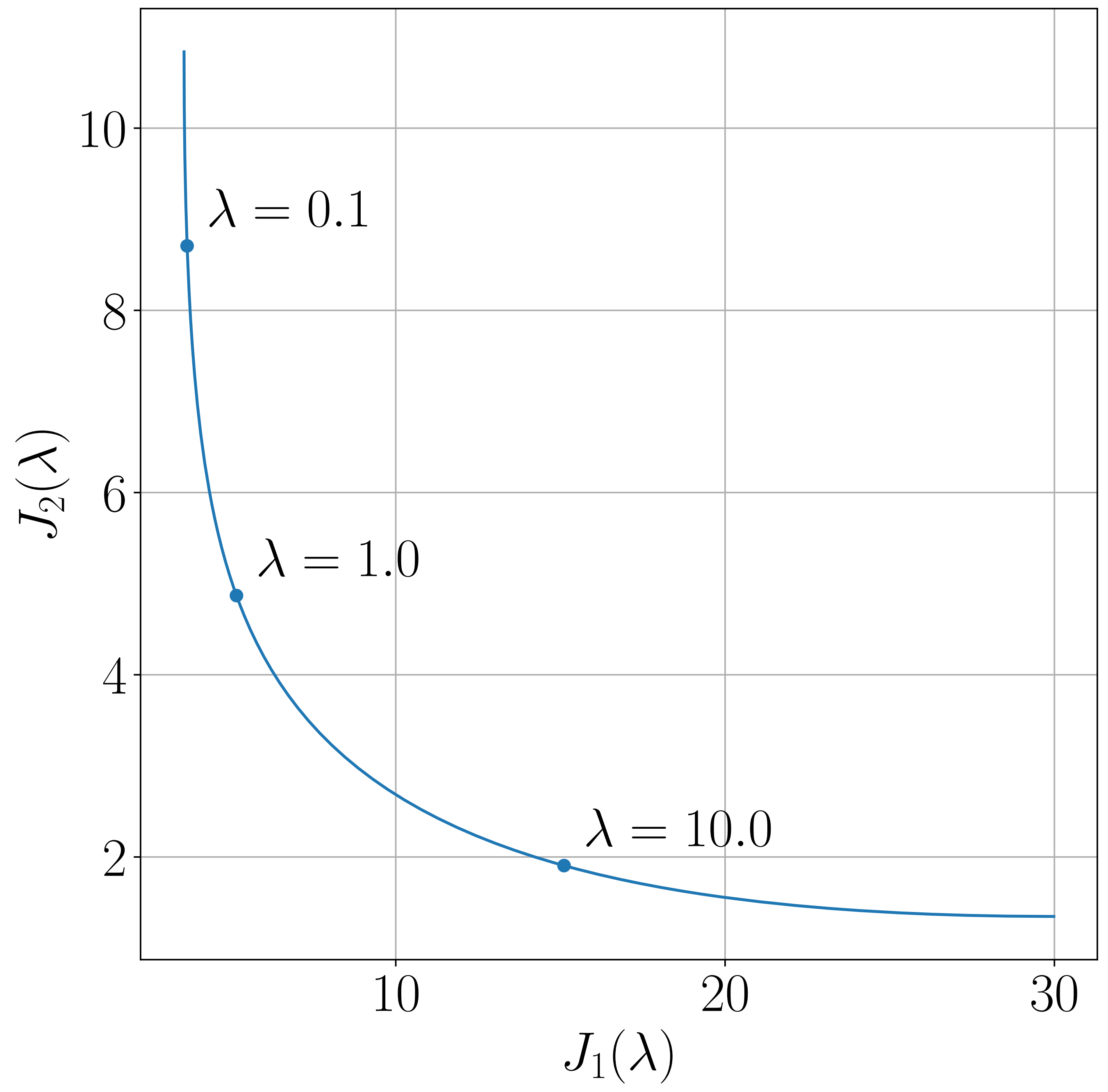
Example

minimize $\quad J_1(x) + \lambda J_2(x)$ $\qquad$ (A_1, A_2 are both 10×5)





Trade-off curve



Using multi-objective least squares

1. Identify **primary objective**
   basic quantity to minimize

2. Choose one or more **secondary objectives**
   quantities that we would like to be small, if possible
   (e.g., size of x, roughness of x, distance from give point)

3. Tweak/tune weights until we like $x^\star(\lambda)$

Using multi-objective least squares

1. Identify **primary objective**
   basic quantity to minimize

2. Choose one or more **secondary objectives**
   quantities that we would like to be small, if possible
   (e.g., size of x, roughness of x, distance from give point)

3. Tweak/tune weights until we like $x^\star(\lambda)$

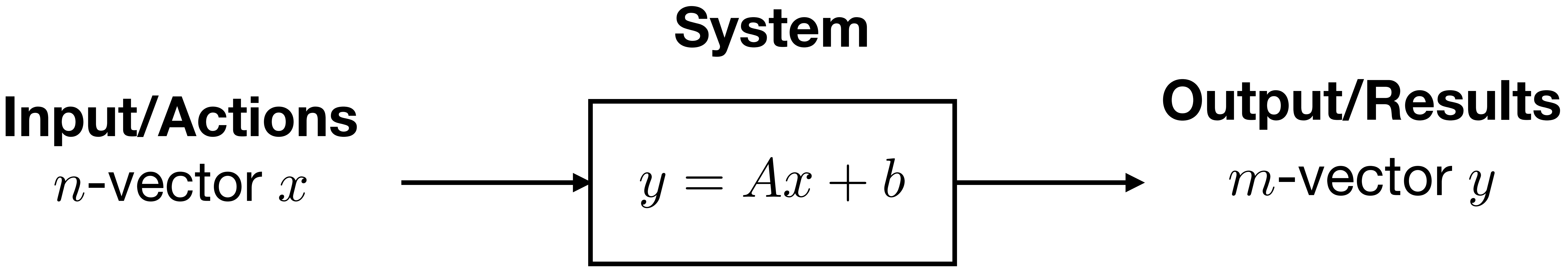
Bi-criterion problem

$$\text{minimize} \quad J_1(x) + \lambda J_2(x)$$

- If J_2 too big, increase λ
- If J_1 too big, decrease λ

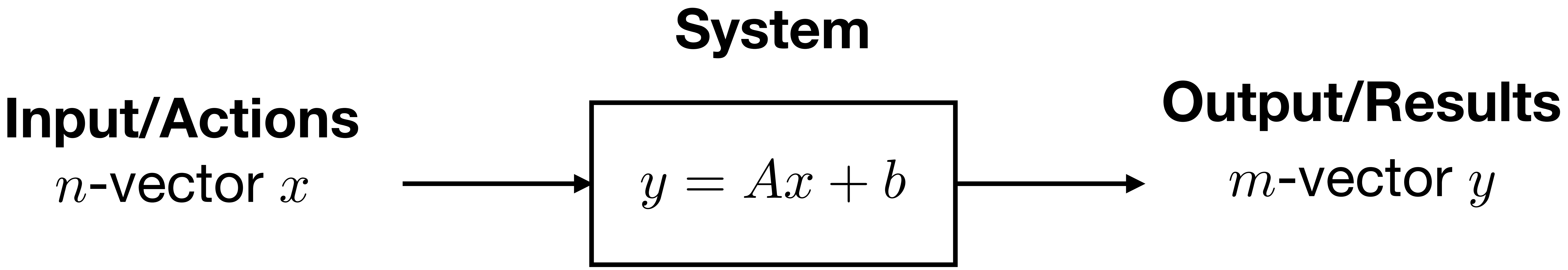
Control

Control



A and b are the known *input-output mapping* of the system.
(analytical models, data fitting, etc.)

Control

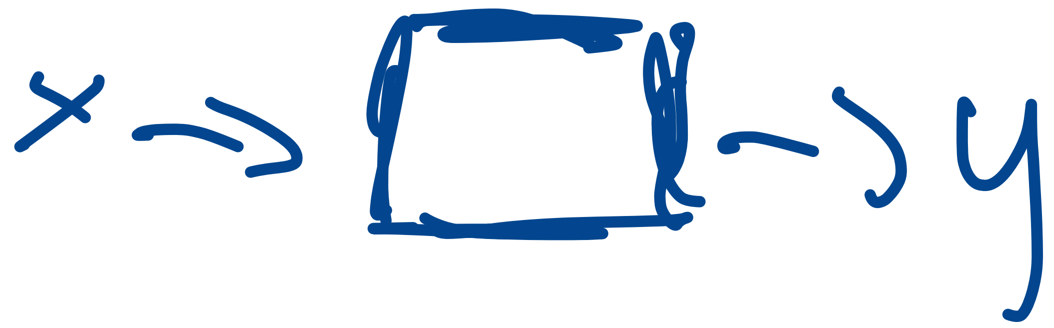


A and b are the known *input-output mapping* of the system.
(analytical models, data fitting, etc.)

Goal

Choose x (which determines y) to optimize
multiple objectives of x and y

Multi-objective control



Optimization problem

minimize $\quad J_1(x) + \lambda J_2(x)$

$J_2 = \|A_2 x - b_2\|^2$

$A_2 = I$

$b_2 = 0$

Primary objective

$$J_1 = \|y - y^{\mathrm{des}}\|^2$$

desired output

Secondary objective

- $J_2 = \|x\|^2$
  (make x small)
- $J_2 = \|x - x^{\mathrm{nom}}\|^2$
  (x close to nominal input)

Product demand shaping

Given n-products,
induce change in demands, n-vector δ^{dem},
by adjusting prices, n-vector δ^{price},

$$\delta^{\text{dem}} = E^{\text{d}} \delta^{\text{price}}$$

Product demand shaping

Given n-products,
induce change in demands, n-vector δ^{dem},
by adjusting prices, n-vector δ^{price},

$$\delta^{\mathrm{dem}} = E^{\mathrm{d}} \delta^{\mathrm{price}}$$

price elasticity of demand matrix

Product demand shaping

Given n-products,
induce change in demands, n-vector δ^{dem},
by adjusting prices, n-vector δ^{price},

$$\delta^{\mathrm{dem}} = E^{\mathrm{d}} \delta^{\mathrm{price}}$$

price elasticity of demand matrix

example E^{d}

$$E^{\mathrm{d}} = \begin{bmatrix} -0.4 & * & * \\ 0.2 & * & * \\ * & * & * \end{bmatrix}$$

$\delta_1^{\mathrm{price}} = 0.01$
(first price $+1\%$)

→

- $\delta_1^{\mathrm{dem}} = -0.004$
  (first demand: -0.4%)
- $\delta_2^{\mathrm{dem}} = 0.002$
  (second demand: $+0.2\%$)

Product demand shaping

System

$$\delta^{\mathrm{dem}} = E^{\mathrm{d}}\delta^{\mathrm{price}}$$

Optimization problem

$$\text{minimize} \quad J_1(x) + \lambda J_2(x)$$

Product demand shaping

System

$$\delta^{\text{dem}} = E^{\text{d}}\delta^{\text{price}}$$

Optimization problem

minimize $\quad J_1(x) + \lambda J_2(x)$

Primary objective

$$\begin{aligned} J_1 &= \|\delta^{\text{dem}} - \delta^{\text{tar}}\|^2 \\ &= \|E^{\text{d}}\delta^{\text{price}} - \delta^{\text{tar}}\|^2 \end{aligned}$$

Product demand shaping

System

$$\delta^{\mathrm{dem}} = E^{\mathrm{d}} \delta^{\mathrm{price}}$$

Optimization problem

minimize $\quad J_1(x) + \lambda J_2(x)$

Primary objective

$$\begin{aligned} J_1 &= \|\delta^{\mathrm{dem}} - \delta^{\mathrm{tar}}\|^2 \\ &= \|E^{\mathrm{d}} \delta^{\mathrm{price}} - \delta^{\mathrm{tar}}\|^2 \end{aligned}$$

target demand

Product demand shaping

System

$$\delta^{\text{dem}} = E^{\text{d}} \delta^{\text{price}}$$

Optimization problem

$$\text{minimize} \quad J_1(x) + \lambda J_2(x)$$

Primary objective

$$\begin{aligned} J_1 &= \|\delta^{\text{dem}} - \delta^{\text{tar}}\|^2 \\ &= \|E^{\text{d}} \delta^{\text{price}} - \delta^{\text{tar}}\|^2 \end{aligned}$$

target demand

Secondary objective

$$J_2 = \|\delta^{\text{price}}\|^2$$

Product demand shaping

System

$$\delta^{\text{dem}} = E^{\text{d}} \delta^{\text{price}}$$

Optimization problem

minimize $\quad J_1(x) + \lambda J_2(x)$

Primary objective

$$J_1 = \|\delta^{\text{dem}} - \delta^{\text{tar}}\|^2$$
$$= \|E^{\text{d}} \delta^{\text{price}} - \delta^{\text{tar}}\|^2$$

target demand

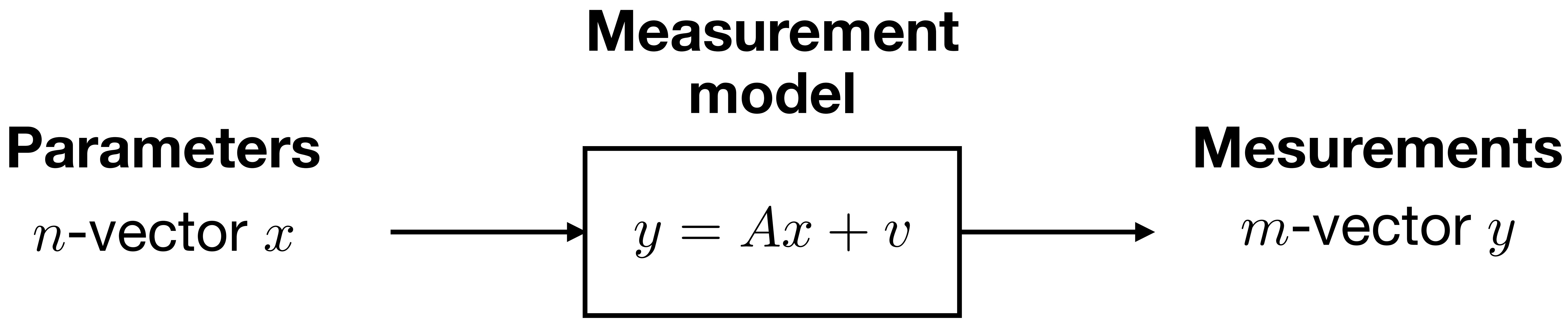
Secondary objective

$$J_2 = \|\delta^{\text{price}}\|^2$$

don't change prices too much

Estimation and inversion

Estimation



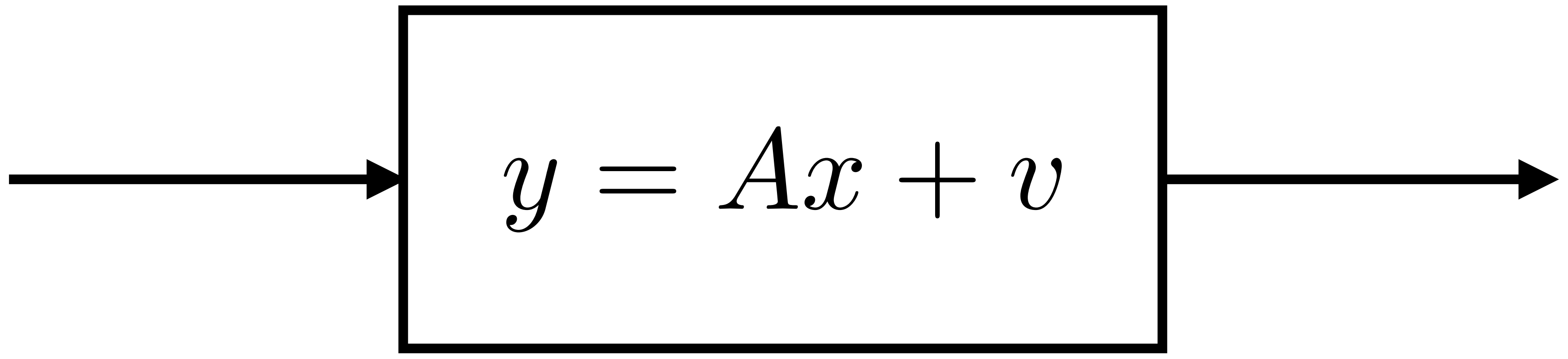
m-vector v are (unknown) *noises* or *measurement errors*

Estimation

Measurement model

Parameters

n-vector x



Mesurements

m-vector y

m-vector v are (unknown) *noises* or *measurement errors*

Basic least squares estimation

(assuming v is small and A as independent columns)

minimize $\quad J_1 = \|Ax - y\|^2$

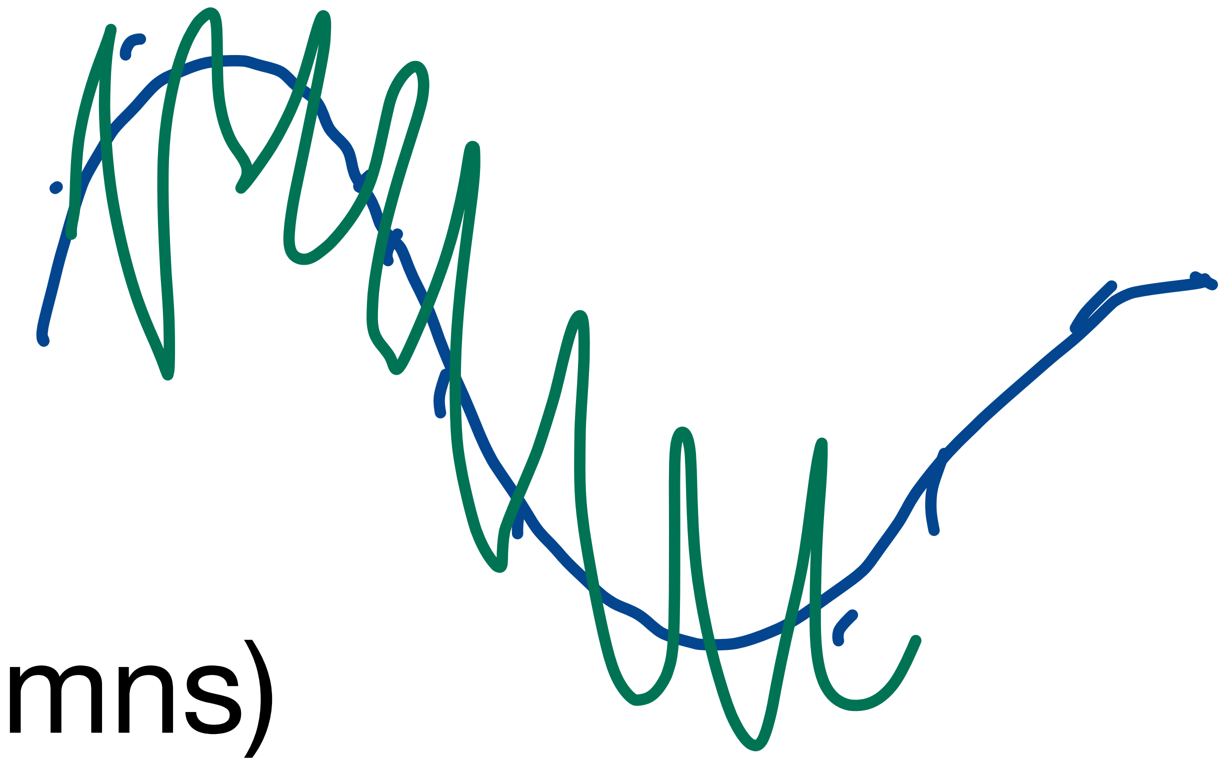
Regularized inversion

Basic least squares estimation

(assuming v is small and A as independent columns)

minimize $\quad J_1 = \|Ax - y\|^2$

Regularized inversion



Basic least squares estimation

(assuming v is small and A as independent columns)

$$\text{minimize} \quad J_1 = \|Ax - y\|^2$$

Regularization

We can get much better results by incorporating prior information about x

- x small: $J_2 = \|x\|^2$ ("Tikhonov regularization")
- x is smooth: $J_2 = \|Dx\|^2 = \sum_{i=1}^{n-1}(x_{i+1} - x_i)^2$
- x close to prior: $J_2 = \|x - x^{\text{prior}}\|^2$

$$D = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -1 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & -1 & 1 \end{bmatrix}$$

Regularized inversion

Optimization problem

$$\text{minimize} \quad J_1(x) + \lambda J_2(x)$$

- Adjust λ until you are happy with the results
- curve $x^\star(\lambda)$ is the *regularization path*

Regularized inversion

Optimization problem

$$\text{minimize} \quad J_1(x) + \lambda J_2(x)$$

- Adjust λ until you are happy with the results
- curve $x^\star(\lambda)$ is the *regularization path*

Example Tikhonov regularization

$$\text{minimize} \quad \|Ax - y\|^2 + \lambda \|x\|^2 = \|\tilde{A}x - \tilde{b}\|^2$$

$$\tilde{A} = \begin{bmatrix} A \\ \sqrt{\lambda} I \end{bmatrix}, \qquad \tilde{b} = \begin{bmatrix} y \\ 0 \end{bmatrix}$$

Regularized inversion

Optimization problem

$$\text{minimize} \quad J_1(x) + \lambda J_2(x)$$

- Adjust λ until you are happy with the results
- curve $x^\star(\lambda)$ is the *regularization path*

Example Tikhonov regularization

$$\text{minimize} \quad \|Ax - y\|^2 + \lambda\|x\|^2 = \|\tilde{A}x - \tilde{b}\|^2$$

$$\tilde{A} = \begin{bmatrix} A \\ \sqrt{\lambda}I \end{bmatrix}, \qquad \tilde{b} = \begin{bmatrix} y \\ 0 \end{bmatrix}$$

$\tilde{A}$ has always linearly independent columns

$\tilde{A}x = (Ax, \sqrt{\lambda}x) = 0$ if and only if $\sqrt{\lambda}x = 0 \Rightarrow x = 0$

Images representation

Monochrome images

Images represented as an $m \times n$ matrix X

Each value X_{ij} represents a pixel's intensity ($0 =$ black, $1 =$ white)
(sometimes $0 =$ black, and $255 =$ white)

We can represent an $m \times n$ matrix X by a single vector $x \in \mathbf{R}^{mn}$

$$X_{ij} = x_k, \qquad k = m(j-1) + i$$

Monochrome image

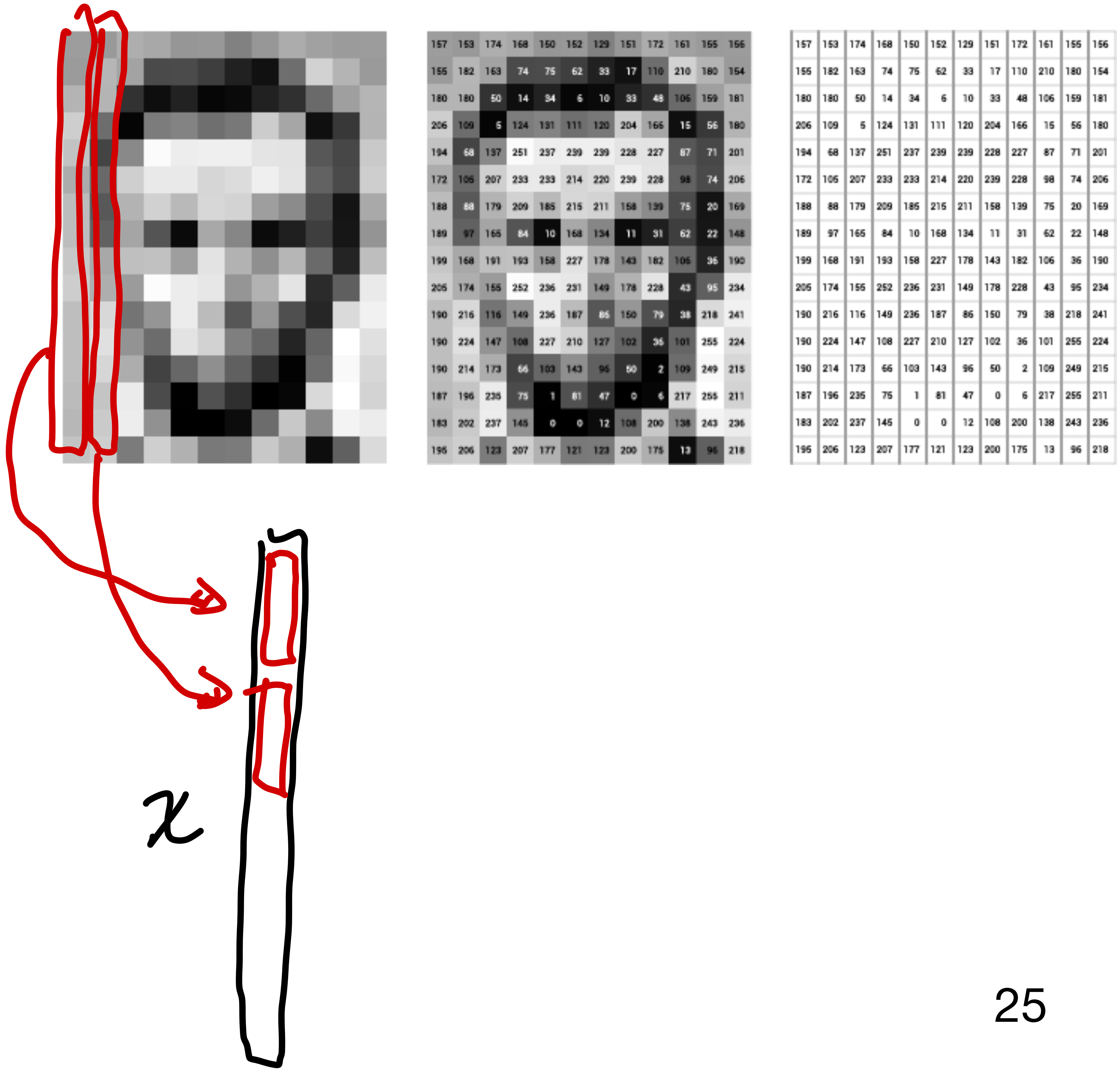


Image de-blurring

Given a noisy **blurred image** y (vector form of Y)

Model

$$y = Ax + v$$

(**blurring matrix** A,
i.e., convolution)

Image de-blurring

Given a noisy **blurred image** y (vector form of Y)

Model

$$y = Ax + v$$

(**blurring matrix** A, i.e., convolution)

Least-squares de-blurring

Find x (vector form of X) by solving

$$\text{minimize} \quad \|Ax - y\|^2 + \lambda \left(\|D_v x\|^2 + \|D_h x\|^2\right)$$

Image de-blurring

Given a noisy **blurred image** y (vector form of Y)

Model

$$y = Ax + v$$

(**blurring matrix** A, i.e., convolution)

Least-squares de-blurring

Find x (vector form of X) by solving

$$\text{minimize} \quad \|Ax - y\|^2 + \lambda \left(\|D_v x\|^2 + \|D_h x\|^2 \right)$$

Smoothing regularization with weight λ

Image de-blurring

Given a noisy **blurred image** y (vector form of Y)

Model
$$y = Ax + v$$

(blurring matrix A**,**
i.e., convolution)

Least-squares de-blurring

Find x (vector form of X) by solving

$$\text{minimize} \quad \|Ax - y\|^2 + \lambda \left(\|D_v x\|^2 + \|D_h x\|^2 \right)$$

Smoothing regularization
with weight λ

Vertical differences
$$\sum_i \sum_j (X_{i,j+1} - X_{ij})^2$$

Image de-blurring

Given a noisy **blurred image** y (vector form of Y)

Model

$$y = Ax + v$$

(**blurring matrix** A, i.e., convolution)

Least-squares de-blurring

Find x (vector form of X) by solving

$$\text{minimize} \quad \|Ax - y\|^2 + \lambda \left(\|D_v x\|^2 + \|D_h x\|^2 \right)$$

Smoothing regularization with weight λ

Vertical differences

$$\sum_i \sum_j (X_{i,j+1} - X_{ij})^2$$

Horizontal differences

$$\sum_i \sum_j (X_{i+1,j} - X_{ij})^2$$

Example

Blurred image

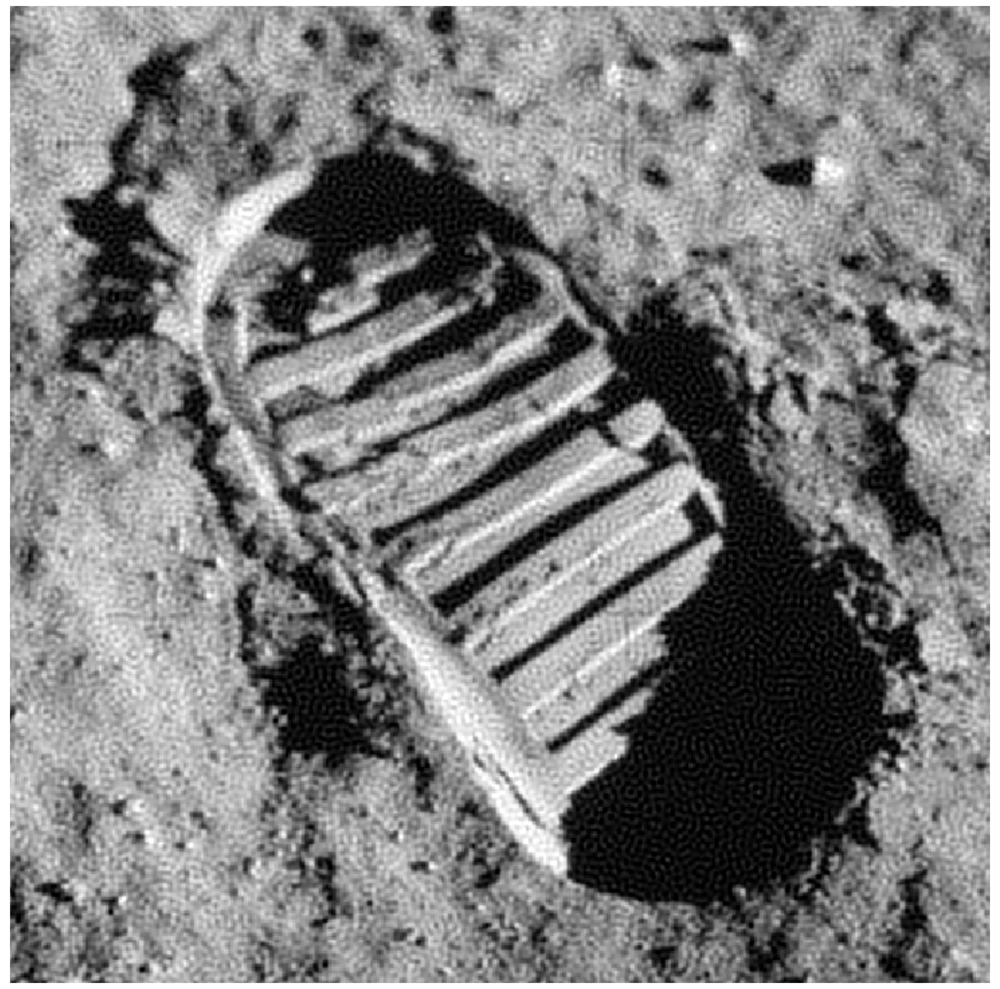
[VMLS book, Page 332]

Example

Blurred image

Regularization path

$\lambda = 10^{-6}$



$\lambda = 10^{-4}$

$\lambda = 10^{-2}$

$\lambda = 10^{-1}$

[VMLS book, Page 332]

Regularized data fitting

Motivation for regularization

Consider the data fitting model (of $y \approx f(x)$)

$$\hat{f}(x) = \theta_1 f_1(x) + \cdots + \theta_p f_p(x)$$

with $f_1(x) = 1$

Motivation for regularization

Consider the data fitting model (of $y \approx f(x)$)

$$\hat{f}(x) = \theta_1 f_1(x) + \cdots + \theta_p f_p(x)$$

with $f_1(x) = 1$

θ_i is the **sensitivity** of $\hat{f}(x)$ to $f_i(x)$ ⟶ It cannot be too large!

Motivation for regularization

Consider the data fitting model (of $y \approx f(x)$)

$$\hat{f}(x) = \theta_1 f_1(x) + \cdots + \theta_p f_p(x)$$

with $f_1(x) = 1$

θ_i is the **sensitivity** of $\hat{f}(x)$ to $f_i(x)$ ⟶ It cannot be too large!

Therefore, we want to **make** $\theta_2, \ldots, \theta_p$ **small**

(θ_1 is an exception, since $f_1(x) = 1$ never changes)

Regularized data fitting

Suppose we have training data

n-vectors $x^{(1)}, \dots, x^{(N)}$, and scalars $y^{(1)}, \dots, y^{(N)}$

We can express the training error as

$$A\theta - y$$

Regularized data fitting

Suppose we have training data

n-vectors $x^{(1)}, \dots, x^{(N)}$, and scalars $y^{(1)}, \dots, y^{(N)}$

We can express the training error as

$$A\theta - y$$

Regularized data fitting

minimize $\|A\theta - y\|^2 + \lambda \|\theta_{2:p}\|^2$

Regularized data fitting

Suppose we have training data

n-vectors $x^{(1)}, \ldots, x^{(N)}$, and scalars $y^{(1)}, \ldots, y^{(N)}$

We can express the training error as

$$A\theta - y$$

Ridge regression

$$\hat{y}^{(i)} = (x^{(i)})^T\beta + v \longrightarrow \hat{y} = X^T\beta + v\mathbf{1}$$

Regularized data fitting

minimize $\|A\theta - y\|^2 + \lambda\|\theta_{2:p}\|^2$ $\longleftrightarrow$ minimize $\|X^T\beta + v\mathbf{1} - y\|^2 + \lambda\|\beta\|^2$

Regularized data fitting

Suppose we have training data

n-vectors $x^{(1)}, \dots, x^{(N)}$, and scalars $y^{(1)}, \dots, y^{(N)}$

We can express the training error as

$$A\theta - y$$

Ridge regression

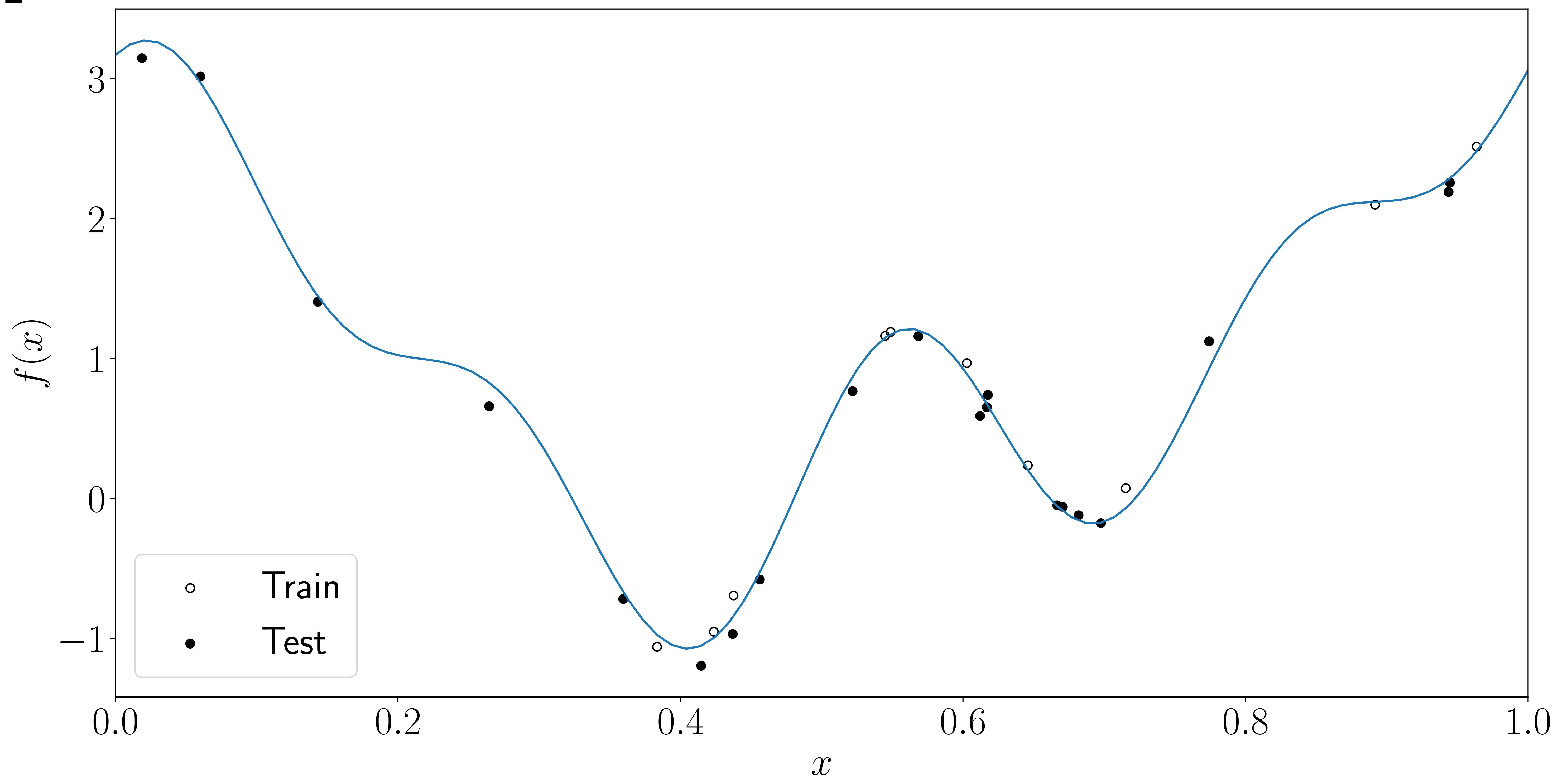
$$\hat{y}^{(i)} = (x^{(i)})^T\beta + v \longrightarrow \hat{y} = X^T\beta + v\mathbf{1}$$

Regularized data fitting

minimize $\|A\theta - y\|^2 + \lambda\|\theta_{2:p}\|^2$ $\longleftrightarrow$ minimize $\|X^T\beta + v\mathbf{1} - y\|^2 + \lambda\|\beta\|^2$

Choose λ with **validation**

Example

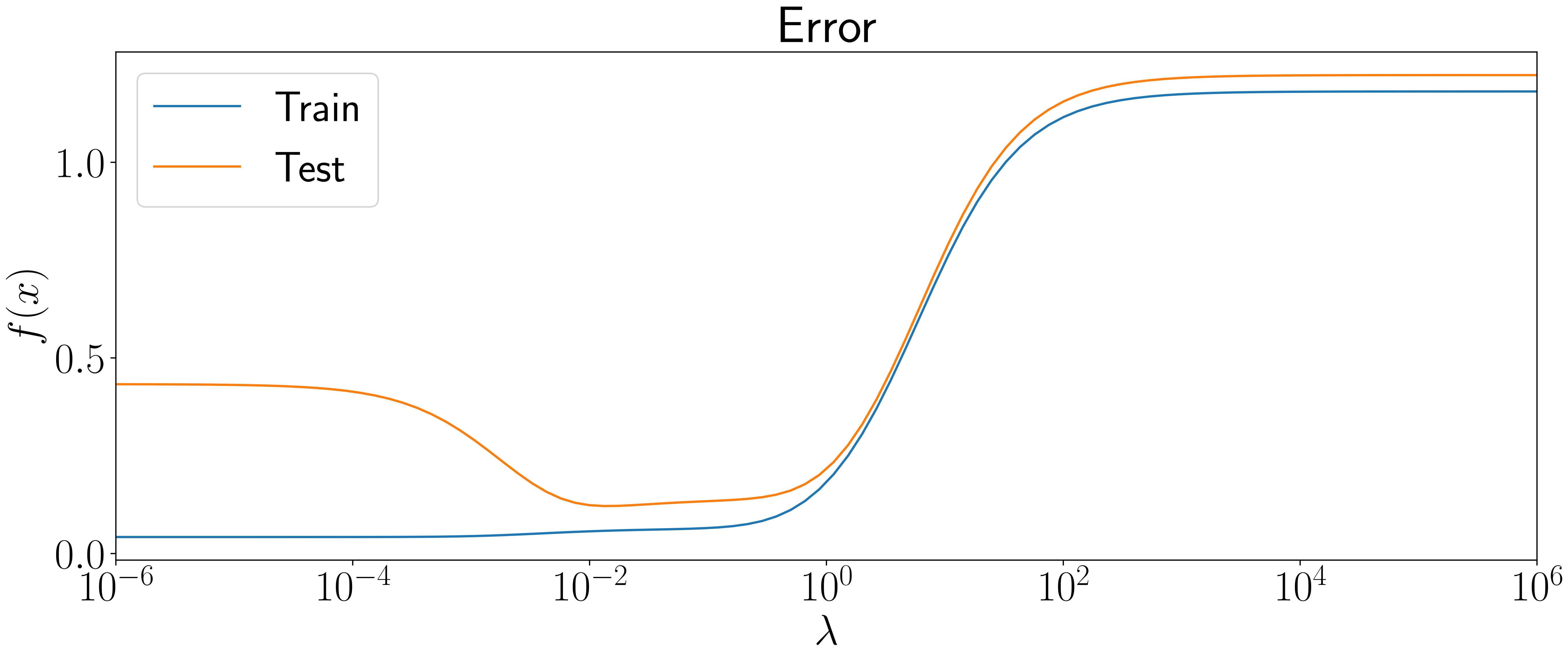


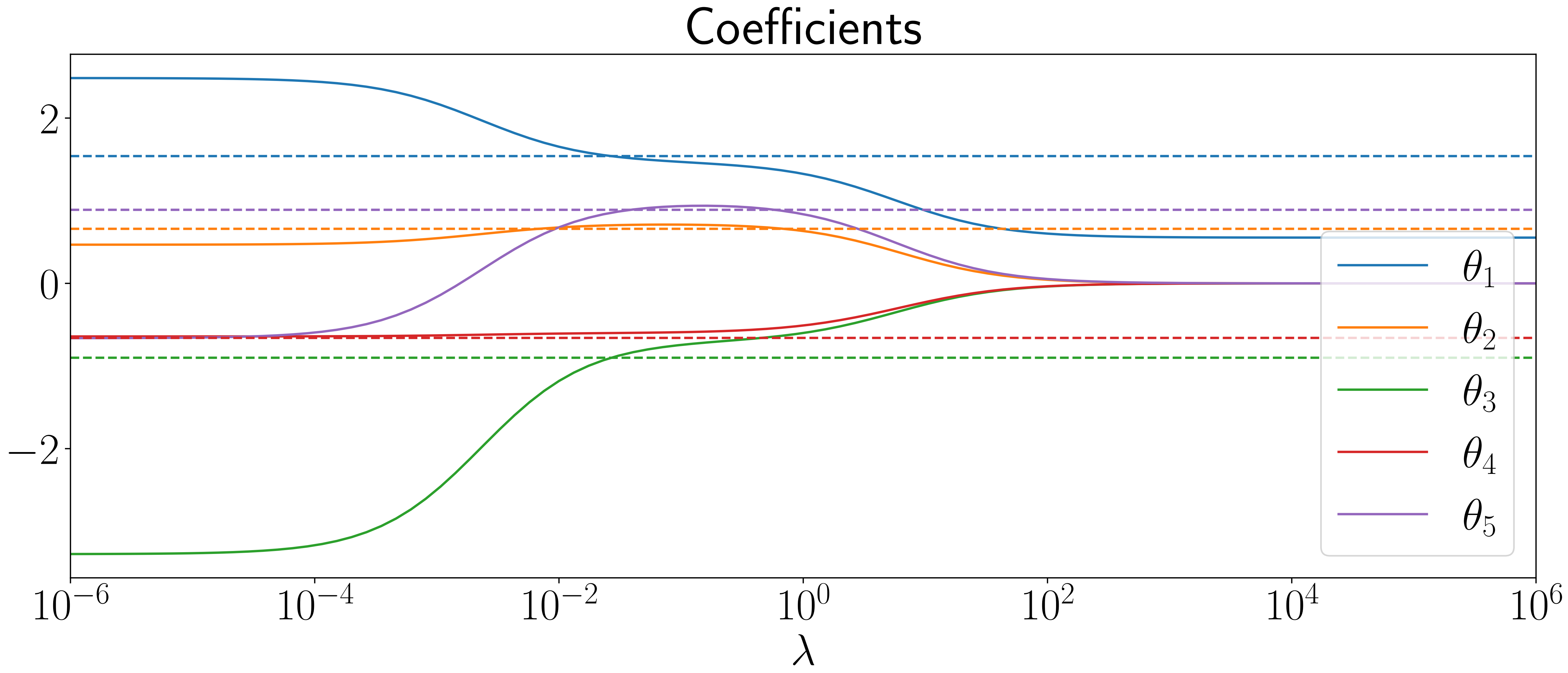
- Solid line to generate synthetic (simulated data)
- Fit a model with 5 parameters $\theta_1, \ldots, \theta_5$

$$\hat{f}(x) = \theta_1 + \sum_{k=1}^{4} \theta_{k+1} \sin(w_k x + \phi_k), \qquad \text{with given } w_k, \phi_k$$

Train and test errors across regularization

- Minimum test error $\lambda \approx 0.013$
- Dashed lines: coefficients to generate data
- For $\lambda \approx 0.013$, estimated coefficients close to true values
- $\theta_{2:5} \to 0$ as $\lambda \to \infty$





Multi-objective least-squares

Today, we learned to:

- **Recognize** and **write** multi-objective least squares problems
- **Solve** multi-objective least squares problems
- **Add regularization** to improve solutions performance

References

- S. Boyd, L. Vandenberghe: Introduction to Applied Linear Algebra – Vectors, Matrices, and Least Squares
  - Chapter 15: multi-objective least squares

Next lecture

- Constrained least squares