10. Multi-objective least squares

- multi-objective least squares
- regularized data fitting
- control
- estimation and inversion
Multi-objective least squares

we have several objectives

\[ J_1 = \|A_1x - b_1\|^2, \ldots, J_k = \|A_kx - b_k\|^2 \]

- \( A_i \) is an \( m_i \times n \) matrix, \( b_i \) is an \( m_i \)-vector
- we seek one \( x \) that makes all \( k \) objectives small
- usually there is a trade-off: no single \( x \) minimizes all objectives simultaneously

**Weighted least squares formulation**: find \( x \) that minimizes

\[ \lambda_1 \|A_1x - b_1\|^2 + \cdots + \lambda_k \|A_kx - b_k\|^2 \]

- coefficients \( \lambda_1, \ldots, \lambda_k \) are positive weights
- weights \( \lambda_i \) express relative importance of different objectives
- without loss of generality, we can choose \( \lambda_1 = 1 \)
Solution of weighted least squares

- weighted least squares is equivalent to a standard least squares problem

\[ \minimize \left\| \begin{bmatrix} \sqrt{\lambda_1} A_1 \\ \sqrt{\lambda_2} A_2 \\ \vdots \\ \sqrt{\lambda_k} A_k \end{bmatrix} x - \begin{bmatrix} \sqrt{\lambda_1} b_1 \\ \sqrt{\lambda_2} b_2 \\ \vdots \\ \sqrt{\lambda_k} b_k \end{bmatrix} \right\|^2 \]

- solution is unique if the *stacked matrix* has linearly independent columns
- each matrix \( A_i \) may have linearly dependent columns (or be a wide matrix)
- if the stacked matrix has linearly independent columns, the solution is

\[ \hat{x} = \left( \lambda_1 A_1^T A_1 + \cdots + \lambda_k A_k^T A_k \right)^{-1} \left( \lambda_1 A_1^T b_1 + \cdots + \lambda_k A_k^T b_k \right) \]
Example with two objectives

minimize \[ \|A_1x - b_1\|^2 + \lambda \|A_2x - b_2\|^2 \]

\(A_1\) and \(A_2\) are 10 \(\times\) 5

plot shows weighted least squares solution \(\hat{x}(\lambda)\) as function of weight \(\lambda\)
Example with two objectives

\[
\text{minimize } \|A_1x - b_1\|^2 + \lambda \|A_2x - b_2\|^2
\]

- left figure shows $J_1(\lambda) = \|A_1\hat{x}(\lambda) - b_1\|^2$ and $J_2(\lambda) = \|A_2\hat{x}(\lambda) - b_2\|^2$
- right figure shows optimal trade-off curve of $J_2(\lambda)$ versus $J_1(\lambda)$
Outline

- multi-objective least squares
- regularized data fitting
- control
- estimation and inversion
Motivation

• consider linear-in-parameters model

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

we assume \( f_1(x) \) is the constant function 1

• we fit the model \( \hat{f}(x) \) to examples \((x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})\)

• large coefficient \( \theta_i \) makes model more sensitive to changes in \( f_i(x) \)

• keeping \( \theta_2, \ldots, \theta_p \) small helps avoid over-fitting

• this leads to two objectives:

\[
J_1(\theta) = \sum_{k=1}^{N} (\hat{f}(x^{(k)}) - y^{(k)})^2, \quad J_2(\theta) = \sum_{j=2}^{p} \theta_j^2
\]

primary objective \( J_1(\theta) \) is sum of squares of prediction errors
Weighted least squares formulation

\[
\text{minimize} \quad J_1(\theta) + \lambda J_2(\theta) = \sum_{k=1}^{N} (\hat{f}(x^{(k)}) - y^{(k)})^2 + \lambda \sum_{j=2}^{p} \theta_j^2
\]

1. \( \lambda \) is positive regularization parameter
2. equivalent to least squares problem: minimize

\[
\left\| \begin{bmatrix} A_1 \\ \sqrt{\lambda} A_2 \end{bmatrix} \theta - \begin{bmatrix} y^d \\ 0 \end{bmatrix} \right\|^2
\]

with \( y^d = (y^{(1)}, \ldots, y^{(N)}) \),

\[
A_1 = \begin{bmatrix}
1 & f_2(x^{(1)}) & \cdots & f_p(x^{(1)}) \\
1 & f_2(x^{(2)}) & \cdots & f_p(x^{(2)}) \\
\vdots & \vdots & \ddots & \vdots \\
1 & f_2(x^{(N)}) & \cdots & f_p(x^{(N)})
\end{bmatrix}, \quad A_2 = \begin{bmatrix}
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 1
\end{bmatrix}
\]

1. stacked matrix has linearly independent columns (for positive \( \lambda \))
2. value of \( \lambda \) can be chosen by out-of-sample validation or cross-validation
Example

- solid line is signal used to generate synthetic (simulated) data
- 10 blue points are used as training set; 20 red points are used as test set
- we fit a model with five parameters $\theta_1, \ldots, \theta_5$:

$$\hat{f}(x) = \theta_1 + \sum_{k=1}^{4} \theta_{k+1} \sin(\omega_k x + \phi_k) \quad \text{(with given } \omega_k, \phi_k)$$
• minimum test RMS error is for $\lambda$ around 0.08
• increasing $\lambda$ “shrinks” the coefficients $\theta_2, \ldots, \theta_5$
• dashed lines show coefficients used to generate the data
• for $\lambda$ near 0.08, estimated coefficients are close to these “true” values
Outline

- multi-objective least squares
- regularized data fitting
- control
- estimation and inversion
Control

\[ y = Ax + b \]

- \( x \) is \( n \)-vector of actions or inputs
- \( y \) is \( m \)-vector of results or outputs
- relation between inputs and outputs is a known affine function

the goal is to choose inputs \( x \) to optimize different objectives on \( x \) and \( y \)
Optimal input design

Linear dynamical system

\[ y(t) = h_0 u(t) + h_1 u(t - 1) + h_2 u(t - 2) + \cdots + h_t u(0) \]

- output \( y(t) \) and input \( u(t) \) are scalar
- we assume input \( u(t) \) is zero for \( t < 0 \)
- coefficients \( h_0, h_1, \ldots \) are the impulse response coefficients
- output is convolution of input with impulse response

Optimal input design

- optimization variable is the input sequence \( x = (u(0), u(1), \ldots, u(N)) \)
- goal is to track a desired output using a small and slowly varying input
Input design objectives

minimize \( J_t(x) + \lambda_v J_v(x) + \lambda_m J_m(x) \)

- primary objective: track desired output \( y_{\text{des}} \) over an interval \([0, N]\):
  \[
  J_t(x) = \sum_{t=0}^{N} (y(t) - y_{\text{des}}(t))^2
  \]

- secondary objectives: use a small and slowly varying input signal:
  \[
  J_m(x) = \sum_{t=0}^{N} u(t)^2, \quad J_v(x) = \sum_{t=0}^{N-1} (u(t + 1) - u(t))^2
  \]
Tracking error

\[ J_t(x) = \sum_{t=0}^{N} (y(t) - y_{\text{des}}(t))^2 = \|A_t x - b_t\|^2 \]

with

\[ A_t = \begin{bmatrix}
  h_0 & 0 & 0 & \cdots & 0 & 0 \\
  h_1 & h_0 & 0 & \cdots & 0 & 0 \\
  h_2 & h_1 & h_0 & \cdots & 0 & 0 \\
  \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
  h_{N-1} & h_{N-2} & h_{N-3} & \cdots & h_0 & 0 \\
  h_N & h_{N-1} & h_{N-2} & \cdots & h_1 & h_0
\end{bmatrix} \]

\[ b_t = \begin{bmatrix}
  y_{\text{des}}(0) \\
  y_{\text{des}}(1) \\
  y_{\text{des}}(2) \\
  \vdots \\
  y_{\text{des}}(N - 1) \\
  y_{\text{des}}(N)
\end{bmatrix} \]
Input variation and magnitude

Input variation

\[
J_v(x) = \sum_{t=0}^{N-1} (u(t + 1) - u(t))^2 = \|Dx\|^2
\]

with \( D \) the \( N \times (N + 1) \) matrix

\[
D = \begin{bmatrix}
-1 & 1 & 0 & \cdots & 0 & 0 & 0 \\
0 & -1 & 1 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & -1 & 1 & 0 \\
0 & 0 & 0 & \cdots & 0 & -1 & 1
\end{bmatrix}
\]

Input magnitude

\[
J_m(x) = \sum_{t=0}^{N} u(t)^2 = \|x\|^2
\]
\( \lambda_v = 0 \), small \( \lambda_m \)

larger \( \lambda_v \) larger \( \lambda_m \)
Outline

- multi-objective least squares
- regularized data fitting
- control
- estimation and inversion
Estimation

Linear measurement model

\[ y = A x_{\text{ex}} + v \]

- \( n \)-vector \( x_{\text{ex}} \) contains parameters that we want to estimate
- \( m \)-vector \( v \) is unknown measurement error or noise
- \( m \)-vector \( y \) contains measurements
- \( m \times n \) matrix \( A \) relates measurements and parameters

**Least squares estimate:** use as estimate of \( x_{\text{ex}} \) the solution \( \hat{x} \) of

\[ \text{minimize} \quad \| A x - y \|^2 \]
Regularized estimation

add other terms to $\|Ax - y\|^2$ to include information about parameters

Example: Tikhonov regularization

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

- goal is to make $\|Ax - y\|$ small with small $x$
- equivalent to solving
  $$(A^TA + \lambda I)x = A^Ty$$
- solution is unique (if $\lambda > 0$) even when $A$ has linearly dependent columns
Signal denoising

- observed signal $y$ is $n$-vector
  
  \[ y = x_{ex} + v \]

- $x_{ex}$ is unknown signal

- $v$ is noise

**Least squares denoising:** find estimate $\hat{x}$ by solving

\[
\text{minimize} \quad ||x - y||^2 + \lambda \sum_{i=1}^{n-1} (x_{i+1} - x_i)^2
\]

goal is to find slowly varying signal $\hat{x}$, close to observed signal $y$
Matrix formulation

\[
\text{minimize } \left\| \begin{bmatrix} I \\ \sqrt{\lambda}D \end{bmatrix} x - \begin{bmatrix} y \\ 0 \end{bmatrix} \right\|^2
\]

- \( D \) is \((n - 1) \times n\) finite difference matrix

\[
D = \begin{bmatrix}
-1 & 1 & 0 & \cdots & 0 & 0 & 0 \\
0 & -1 & 1 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & -1 & 1 & 0 \\
0 & 0 & 0 & \cdots & 0 & -1 & 1 \\
\end{bmatrix}
\]

- equivalent to linear equation

\[
(I + \lambda D^T D)x = y
\]
Trade-off

the two objectives $||\hat{x}(\lambda) - y||$ and $||D\hat{x}(\lambda)||$ for varying $\lambda$
Three solutions

\[
\hat{x}(\lambda) \rightarrow y \text{ for } \lambda \rightarrow 0
\]

\[
\hat{x}(\lambda) \rightarrow \text{avg}(y) \mathbf{1} \text{ for } \lambda \rightarrow \infty
\]

\[
\lambda \approx 10^2 \text{ is good compromise}
\]
Image deblurring

\[ y = A x_{\text{ex}} + v \]

- \( x_{\text{ex}} \) is unknown image, \( y \) is observed image
- \( A \) is (known) blurring matrix, \( v \) is (unknown) noise
- images are \( M \times N \), stored as \( MN \)-vectors

blurred, noisy image \( y \)  
deburred image \( \hat{x} \)
Least squares deblurring

\[
\text{minimize } \|Ax - y\|^2 + \lambda (\|Dv x\|^2 + \|D_h x\|^2)
\]

- 1st term is "data fidelity" term: ensures \( A\hat{x} \approx y \)
- 2nd term penalizes differences between values at neighboring pixels

\[
\|D_h x\|^2 + \|D_v x\|^2 = \sum_{i=1}^{M} \sum_{j=1}^{N-1} (X_{i,j+1} - X_{ij})^2 + \sum_{i=1}^{M-1} \sum_{j=1}^{N} (X_{i+1,j} - X_{ij})^2
\]

if \( X \) is the \( M \times N \) image stored in the \( MN \)-vector \( x \)
Differencing operations in matrix notation

suppose $x$ is the $M \times N$ image $X$, stored column-wise as $MN$-vector

$$x = (X_{1:M,1}, X_{1:M,2}, \ldots, X_{1:M,N})$$

- horizontal differencing: $(N - 1) \times N$ block matrix with $M \times M$ blocks

$$D_h = \begin{bmatrix}
-I & I & 0 & \cdots & 0 & 0 & 0 \\
0 & -I & I & \cdots & 0 & 0 & 0 \\
0 & 0 & -I & I & \cdots & 0 & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & -I & I
\end{bmatrix}$$

- vertical differencing: $N \times N$ block matrix with $(M - 1) \times M$ blocks

$$D_v = \begin{bmatrix}
D & 0 & \cdots & 0 \\
0 & D & \cdots & 0 \\
0 & 0 & \cdots & D
\end{bmatrix}, \quad D = \begin{bmatrix}
-1 & 1 & 0 & \cdots & 0 & 0 \\
0 & -1 & 1 & \cdots & 0 & 0 \\
0 & 0 & 0 & \cdots & -1 & 1
\end{bmatrix}$$
Deblurred images

\( \lambda = 10^{-6} \)

\( \lambda = 10^{-4} \)

\( \lambda = 10^{-2} \)

\( \lambda = 1 \)
Tomography

\[ y = A x_{ex} + \nu \]

- \( x_{ex} \) represents values of some quantity in a region of interest of \( n \) voxels (pixels)
- \( Ax \) represents measurements of the integral along lines through the region

\[ (Ax)_i = \sum_{j=1}^{n} A_{ij} x_j \]

\( A_{ij} \) is the length of the intersection of the line in measurement \( i \) with voxel \( j \)

![Diagram showing intersection of lines with voxels]
Tomographic reconstruction

minimize $\|Ax - y\|^2 + \lambda (\|D_v x\|^2 + \|D_h x\|^2)$

$D_v$ and $D_h$ are defined as in image deblurring example on page 10.23

Example

- left: 4000 lines (100 points, 40 lines per point)
- right: object placed in the square region at the center of the picture on the left
- region of interest is divided in 10000 pixels
Regularized least squares reconstruction

\[ \lambda = 10^{-2} \]

\[ \lambda = 10^{-1} \]

\[ \lambda = 1 \]

\[ \lambda = 5 \]

\[ \lambda = 10 \]

\[ \lambda = 100 \]