11. Constrained least squares

- constrained least squares
- solution of least norm problem
- solution of constrained least squares
- linear quadratic control
- linear quadratic estimation
- portfolio optimization

Constrained least squares

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|^2 \\ \text{subject to} & Cx = d \end{array}$$

- A is an $m \times n$ matrix, C is a $p \times n$ matrix, b is an m-vector, d is a p-vector
- $\|Ax - b\|^2$ is the *objective*, $Cx = d$ are the *constraints*
- we make no assumptions about the shape of A
- in most applications $p < n$ and the equation $Cx = d$ is underdetermined
- goal is to find a solution of $Cx = d$ with smallest objective

Solution

- x is *feasible* if $Cx = d$
- $\hat{x}$ is *optimal* or *solution* if it is feasible and

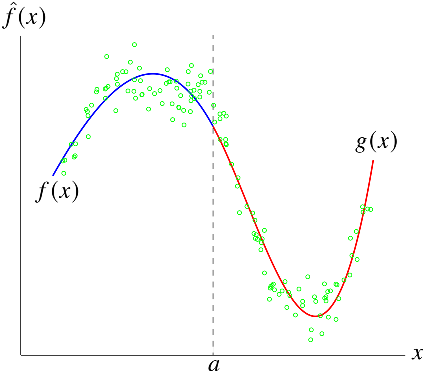
$$\|A\hat{x} - b\|^2 \leq \|Ax - b\|^2 \quad \text{for all feasible } x$$

Example: Piecewise-polynomial fitting

- fit two polynomials $f(x)$, $g(x)$ to points $(x_1, y_1), \ldots, (x_N, y_N)$

$$f(x_i) \approx y_i \text{ for points } x_i \leq a, \quad g(x_i) \approx y_i \text{ for points } x_i > a$$

- make values and derivatives continuous at point a: $f(a) = g(a)$, $f'(a) = g'(a)$



Constrained LS formulation

- assume points are numbered so that $x_1, \ldots, x_M \leq a$ and $x_{M+1}, \ldots, x_N > a$:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{M} \left(f\left(x_i\right) - y_i\right)^2 + \sum_{i=M+1}^{N} \left(g\left(x_i\right) - y_i\right)^2 \\ \text{subject to} & f(a) = g(a), \quad f'(a) = g'(a) \end{array}$$

- for polynomials $f(x) = \theta_1 + \cdots + \theta_d x^{d-1}$ and $g(x) = \theta_{d+1} + \cdots + \theta_{2d} x^{d-1}$

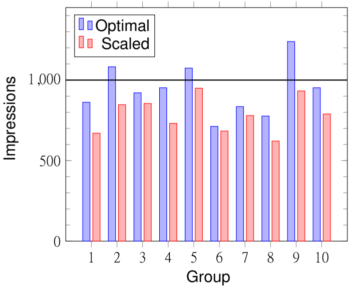
$$A = \begin{bmatrix} 1 & x_1 & \cdots & x_1^{d-1} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 1 & x_M & \cdots & x_M^{d-1} & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & x_{M+1} & \cdots & x_{M+1}^{d-1} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & 1 & x_N & \cdots & x_N^{d-1} \end{bmatrix}, \quad b = \begin{bmatrix} y_1 \\ \vdots \\ y_M \\ y_{M+1} \\ \vdots \\ y_N \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & a & \cdots & a^{d-1} & -1 & -a & \cdots & -a^{d-1} \\ 0 & 1 & \cdots & (d-1)a^{d-2} & 0 & -1 & \cdots & -(d-1)a^{d-2} \end{bmatrix}, \quad d = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Example: Advertising budget allocation

- m demographics groups (audiences), n advertising channels
- v_i^{des} is target number of views or impressions for group i
- s_j is amount of advertising purchased in channel j
- R_{ij} is # views in group i per dollar spent on ads in channel j
- $(Rs)_i$ is total number of views in group i
- fixed budget $\mathbf{1}^T s = B$
- constrained LS problem: minimize $\|Rs - v^{\text{des}}\|^2$ subject to $\mathbf{1}^T s = B$

Example: optimal and scaled LS solution to satisfy budget



Least norm problem

$$\begin{array}{ll} \text{minimize} & \|x\|^2 \\ \text{subject to} & Cx = d \end{array}$$

- C is a $p \times n$ matrix, d is a p-vector
- the goal is to find the solution of $Cx = d$ with the smallest norm
- a special case of constrained LS with $A = I$ and $b = 0$

Least distance problem: minimizing the distance to a given point $a \neq 0$:

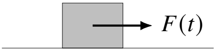
$$\begin{array}{ll} \text{minimize} & \|x - a\|^2 \\ \text{subject to} & Cx = d \end{array}$$

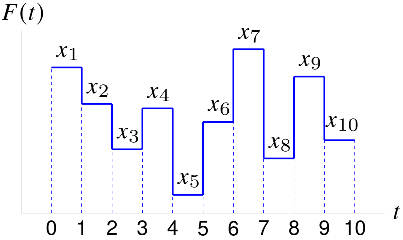
- reduces to least norm problem by a change of variables $y = x - a$

$$\begin{array}{ll} \text{minimize} & \|y\|^2 \\ \text{subject to} & Cy = d - Ca \end{array}$$

- from least norm solution y, we obtain solution $x = y + a$ of first problem

Force sequence





- a unit mass with zero initial position and velocity
- we apply piecewise-constant force $F(t)$ during interval $[0, 10)$:

$$F(t) = x_j \quad \text{for } t \in [j-1, j), \quad j = 1, \ldots, 10$$

- position and velocity at $t = 10$ are given by

$$p^{\text{fin}} = (19/2)x_1 + (17/2)x_2 + (15/2)x_3 + \cdots + (1/2)x_{10}$$

$$v^{\text{fin}} = x_1 + x_2 + \cdots + x_{10}$$

we want to choose a force sequence that results in $p^{\text{fin}} = 1$, $v^{\text{fin}} = 0$

Example

there are many solution; we consider two solutions:

1. *bang-bang force:* solutions with only two nonzero elements:

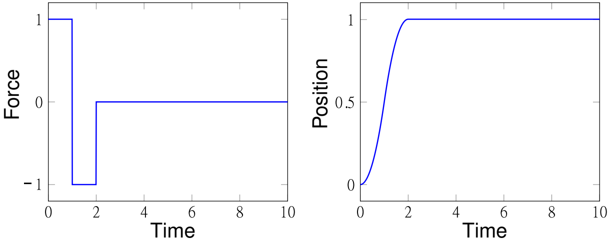
$$x = (1, -1, 0, \ldots, 0), \quad x = (0, 1, -1, \ldots, 0), \quad \ldots$$

2. *least norm solution:* smallest force sequence

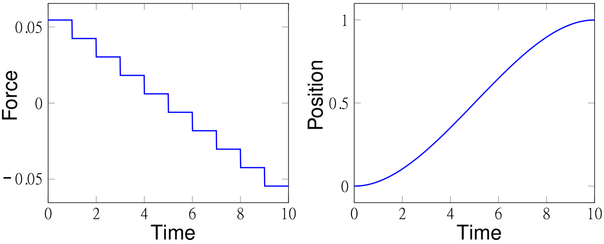
$$\begin{array}{ll} \text{minimize} & \int_0^{10} F(t)^2 dt = \|x\|^2 \\ \text{subject to} & \begin{bmatrix} 19/2 & 17/2 & 15/2 & \cdots & 1/2 \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix} x = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{array}$$

Example results

Bang-bang force



Least norm force



Outline

- constrained least squares
- **solution of least norm problem**
- solution of constrained least squares
- linear quadratic control
- linear quadratic estimation
- portfolio optimization

Solution of least norm problem

$$\begin{array}{ll} \text{minimize} & \|x\|^2 \\ \text{subject to} & Cx = d \end{array}$$

Assumption: we assume that C has linearly independent rows

- $Cx = d$ has at least one solution for every d
- C is wide or square $(p \leq n)$; if $p < n$ there are infinitely many solutions

Solution of least norm problem

$$\hat{x} = C^T(CC^T)^{-1}d$$

- in other words if $Cx = d$ and $x \neq \hat{x}$, then $\|x\| > \|\hat{x}\|$
- unique solution under the above assumption
- $C^T(CC^T)^{-1} = C^\dagger$ is the pseudo-inverse of C, which is also a right-inverse

Proof

1. we first verify that $\hat{x}$ satisfies the constraints:

$$C\hat{x} = CC^T(CC^T)^{-1}d = d$$

2. next we show that $\|x\| > \|\hat{x}\|$ if $Cx = d$ and $x \neq \hat{x}$

$$\begin{aligned} \|x\|^2 &= \|\hat{x} + x - \hat{x}\|^2 \\ &= \|\hat{x}\|^2 + 2\hat{x}^T(x - \hat{x}) + \|x - \hat{x}\|^2 \\ &= \|\hat{x}\|^2 + \|x - \hat{x}\|^2 \\ &\geq \|\hat{x}\|^2 \quad \text{with equality only if } x = \hat{x} \end{aligned}$$

line 3 follows from

$$\hat{x}^T(x - \hat{x}) = d^T(CC^T)^{-1}C(x - \hat{x}) = 0$$

where we used $Cx = C\hat{x} = d$

QR factorization method

using the QR factorization $C^T = QR$ of C^T, we get

$$\begin{aligned}\hat{x} &= C^T(CC^T)^{-1}d \\ &= QR(R^TQ^TQR)^{-1}d \\ &= QR(R^TR)^{-1}d \\ &= QR^{-T}d\end{aligned}$$

Algorithm

1. compute QR factorization $C^T = QR$ ($2p^2n$ flops)
2. solve $R^Tz = d$ by forward substitution (p^2 flops)
3. matrix-vector product $\hat{x} = Qz$ ($2pn$ flops)

complexity: $2p^2n$ flops

Example

$$C = \begin{bmatrix} 1 & -1 & 1 & 1 \\ 1 & 0 & 1/2 & 1/2 \end{bmatrix}, \quad d = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- QR factorization $C^T = QR$

$$\begin{bmatrix} 1 & 1 \\ -1 & 0 \\ 1 & 1/2 \\ 1 & 1/2 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/\sqrt{2} \\ -1/2 & 1/\sqrt{2} \\ 1/2 & 0 \\ 1/2 & 0 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 0 & 1/\sqrt{2} \end{bmatrix}$$

- solve $R^T z = b$

$$\begin{bmatrix} 2 & 0 \\ 1 & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow z_1 = 0, z_2 = \sqrt{2}$$

- evaluate $\hat{x} = Qz = (1, 1, 0, 0)$

Outline

- constrained least squares
- solution of least norm problem
- **solution of constrained least squares**
- linear quadratic control
- linear quadratic estimation
- portfolio optimization

Assumptions

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|^2 \\ \text{subject to} & Cx = d \end{array}$$

Assumptions

1. the stacked $(m + p) \times n$ matrix

$$\begin{bmatrix} A \\ C \end{bmatrix}$$

   has linearly independent columns (left-invertible)

2. $p \times n$ matrix C has linearly independent rows (right-invertible)

assumptions imply that $p \leq n \leq m + p$

Optimality conditions

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|^2 \\ \text{subject to} & Cx = d \end{array}$$

$\hat{x}$ solves the constrained LS problem if and only if there exists a z such that

$$\begin{bmatrix} A^T A & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ z \end{bmatrix} = \begin{bmatrix} A^T b \\ d \end{bmatrix}$$

- this is a set of $n + p$ linear equations in $n + p$ variables
- equations are also known as *Karush-Kuhn-Tucker* (*KKT*) equations
- matrix on left is called *KKT matrix*

Special cases

- least squares: when $p = 0$, reduces to normal equations $A^T A\hat{x} = A^T b$
- least norm: when $A = I$, $b = 0$, reduces to $C\hat{x} = d$ and $\hat{x} + C^T z = 0$

Proof

suppose x satisfies $Cx = d$, and $(\hat{x}, z)$ satisfies optimality conditions, then

$$\begin{aligned}
\|Ax - b\|^2 &= \|A(x - \hat{x}) + A\hat{x} - b\|^2 \\
&= \|A(x - \hat{x})\|^2 + \|A\hat{x} - b\|^2 + 2(x - \hat{x})^T A^T (A\hat{x} - b) \\
&= \|A(x - \hat{x})\|^2 + \|A\hat{x} - b\|^2 - 2(x - \hat{x})^T C^T z \\
&= \|A(x - \hat{x})\|^2 + \|A\hat{x} - b\|^2 \\
&\geq \|A\hat{x} - b\|^2
\end{aligned}$$

- on line 3 we use $A^T A\hat{x} + C^T z = A^T b$; on line $4, Cx = C\hat{x} = d$
- inequality shows that $\hat{x}$ is optimal
- $\hat{x}$ is the unique optimum because equality holds only if

$$A(x - \hat{x}) = 0, \quad C(x - \hat{x}) = 0 \quad \Longrightarrow \quad x = \hat{x}$$

  by the first assumption

Nonsingularity

the KKT matrix

$$\begin{bmatrix} A^T A & C^T \\ C & 0 \end{bmatrix}$$

is nonsingular (invertible) if and only if the two assumptions hold

Proof: if assumptions hold

$$\begin{aligned} \begin{bmatrix} A^T A & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} &\Longrightarrow x^T(A^T A x + C^T z) = 0, \quad Cx = 0 \\ &\Longrightarrow \|Ax\|^2 = 0, \quad Cx = 0 \\ &\Longrightarrow Ax = 0, \quad Cx = 0 \\ &\Longrightarrow x = 0 \quad \text{by assumption } 1 \end{aligned}$$

if $x = 0$, we have $C^T z = -A^T A x = 0$; hence also $z = 0$ by assumption 2

Singularity

if the assumptions do not hold, then the matrix

$$\begin{bmatrix} A^T A & C^T \\ C & 0 \end{bmatrix}$$

is singular

- if assumption 1 does not hold, there exists $x \neq 0$ with $Ax = 0$, $Cx = 0$; then

$$\begin{bmatrix} A^T A & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ 0 \end{bmatrix} = 0$$

- if assumption 2 does not hold there exists a $z \neq 0$ with $C^T z = 0$; then

$$\begin{bmatrix} A^T A & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} 0 \\ z \end{bmatrix} = 0$$

in both cases, this shows that the matrix is singular

Solving KKT equation directly

$$\begin{bmatrix} A^TA & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} A^Tb \\ d \end{bmatrix}$$

Algorithm

1. compute $H = A^TA$ (mn^2 flops)
2. compute $c = A^Tb$ ($2mn$ flops)
3. solve the linear equation

$$\begin{bmatrix} H & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} c \\ d \end{bmatrix}$$

   by the LU factorization ($(2/3)(p+n)^3$ flops) or QR factorization ($2(n+p)^3$)

complexity: $mn^2 + (2/3)(p+n)^3$ flops

Solution by QR factorization

we derive a method that avoid computing gram matrix by using QR factorization

$$\begin{bmatrix} A^TA & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ z \end{bmatrix} = \begin{bmatrix} A^Tb \\ d \end{bmatrix}$$

- multiply 2nd eq. by C^T, add to 1st eq. , make change of variables $w = z - d$,

$$\begin{bmatrix} A^TA + C^TC & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ w \end{bmatrix} = \begin{bmatrix} A^Tb \\ d \end{bmatrix}$$

- assumption 1 guarantees $A^TA + C^TC$ is nonsingular and QR factorization exists:

$$\begin{bmatrix} A \\ C \end{bmatrix} = QR = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R$$

Solution by QR factorization

substituting $A = Q_1 R$ and $C = Q_2 R$ gives the equation

$$\begin{bmatrix} R^T R & R^T Q_2^T \\ Q_2 R & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ w \end{bmatrix} = \begin{bmatrix} R^T Q_1^T b \\ d \end{bmatrix}$$

- multiply first equation with R^{-T} and make change of variables $y = R\hat{x}$

$$\begin{bmatrix} I & Q_2^T \\ Q_2 & 0 \end{bmatrix} \begin{bmatrix} y \\ w \end{bmatrix} = \begin{bmatrix} Q_1^T b \\ d \end{bmatrix}$$

- next we note that the matrix $Q_2 = CR^{-1}$ has linearly independent rows:

$$Q_2^T u = R^{-T} C^T u = 0 \quad \Longrightarrow \quad C^T u = 0 \quad \Longrightarrow \quad u = 0$$

  because C has linearly independent rows (assumption 2)

Solution by QR factorization

we use the QR factorization of Q_2^T to solve

$$\begin{bmatrix} I & Q_2^T \\ Q_2 & 0 \end{bmatrix} \begin{bmatrix} y \\ w \end{bmatrix} = \begin{bmatrix} Q_1^T b \\ d \end{bmatrix}$$

- from the 1st block row, $y = Q_1^T b - Q_2^T w$; substitute this in the 2nd row:

$$Q_2 Q_2^T w = Q_2 Q_1^T b - d$$

- we solve this equation for w using the QR factorization $Q_2^T = \tilde{Q}\tilde{R}$:

$$\tilde{R}^T \tilde{R} w = \tilde{R}^T \tilde{Q}^T Q_1^T b - d$$

  which can be simplified to

$$\tilde{R} w = \tilde{Q}^T Q_1^T b - \tilde{R}^{-T} d$$

  after solving for w, we get $y = Q_1^T b - Q_2^T w$ and solve for $\hat{x}$ in $y = R\hat{x}$

Summary of QR factorization method

$$\begin{bmatrix} A^TA + C^TC & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ w \end{bmatrix} = \begin{bmatrix} A^Tb \\ d \end{bmatrix}$$

Algorithm

1. compute the two QR factorizations

$$\begin{bmatrix} A \\ C \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R \quad \text{and} \quad Q_2^T = \tilde{Q}\tilde{R}$$

2. solve $\tilde{R}^T u = d$ by forward substitution and compute $c = \tilde{Q}^T Q_1^T b - u$
3. solve $\tilde{R} w = c$ by back substitution and compute $y = Q_1^T b - Q_2^T w$
4. compute $R\hat{x} = y$ by back substitution

Complexity

- $2(m+p)n^2 + 2np^2$ flops (QR factorizations dominates)
- order $(m+p)n^2$ due to assumption $p \leq n \leq m+p$

Comparison of the two methods

Complexity: LU is slightly more efficient

- LU factorization

$$mn^2 + \frac{2}{3}(p+n)^3 \leq mn^2 + \frac{16}{3}n^3 \text{ flops}$$

- QR factorization

$$2(p+m)n^2 + 2np^2 \leq 2mn^2 + 4n^3 \text{ flops}$$

upper bounds follow from $p \leq n$ (assumption 2)

Stability

- QR factorization method avoids calculation of Gram matrix A^TA
- hence more robust/stable to numerical errors

Outline

- constrained least squares
- solution of least norm problem
- solution of constrained least squares
- **linear quadratic control**
- linear quadratic estimation
- portfolio optimization

Linear quadratic control

Linear dynamical system

$$x_{t+1} = A_t x_t + B_t u_t, \quad y_t = C_t x_t, \quad t = 1, 2, \ldots$$

- n-vector x_t is system *state* (at time t)
- m-vector u_t is system *input* (we control)
- p-vector y_t is system *output*
- x_t, u_t, y_t are typically desired to be small

Objective: choose inputs $u_1, \ldots, u_{T-1}$ that minimizes $J_{\text{output}} + \rho J_{\text{input}}$ with

$$J_{\text{output}} = \|y_1 - y_1^{\text{des}}\|^2 + \cdots + \|y_T - y_T^{\text{des}}\|^2, \quad J_{\text{input}} = \|u_1\|^2 + \cdots + \|u_{T-1}\|^2$$

where y_i^{des} are given desired values (possibly zero)

Constraints

- dynamics constraint
- initial state and (possibly) the final state are specified $x_1 = x^{\text{init}}$, $x_T = x^{\text{des}}$

Linear quadratic control problem

$$
\begin{array}{ll}
\text{minimize} & \|C_1x_1 - y_1^{\text{des}}\|^2 + \cdots + \|C_Tx_T - y_T^{\text{des}}\|^2 + \rho\left(\|u_1\|^2 + \cdots + \|u_{T-1}\|^2\right) \\
\text{subject to} & x_{t+1} = A_tx_t + B_tu_t, \quad t = 1, \ldots, T-1 \\
& x_1 = x^{\text{init}}, \quad x_T = x^{\text{des}}
\end{array}
$$

variables: $x_1, \ldots, x_T$ and $u_1, \ldots, u_{T-1}$

Constrained least squares formulation

$$
\begin{array}{ll}
\text{minimize} & \|\tilde{A}z - \tilde{b}\|^2 \\
\text{subject to} & \tilde{C}z = \tilde{d}
\end{array}
$$

variables: the $(nT + m(T-1))$-vector

$$
z = (x_1, \ldots, x_T, u_1, \ldots, u_{T-1})
$$

Linear quadratic control problem

Objective function: $\|\tilde{A}z - \tilde{b}\|^2$ with

$$\tilde{A} = \left[\begin{array}{ccc|ccc} C_1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & C_T & 0 & \cdots & 0 \\ \hline 0 & \cdots & 0 & \sqrt{\rho}I & \cdots & 0 \\ \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & \sqrt{\rho}I \end{array}\right], \quad \tilde{b} = \left[\begin{array}{c} y_1^{\mathrm{des}} \\ \vdots \\ y_T^{\mathrm{des}} \\ \hline 0 \\ \vdots \\ 0 \end{array}\right]$$

Constraints: $\tilde{C}z = \tilde{d}$ with

$$\tilde{C} = \left[\begin{array}{cccccc|cccc} A_1 & -I & 0 & \cdots & 0 & 0 & B_1 & 0 & \cdots & 0 \\ 0 & A_2 & -I & \cdots & 0 & 0 & 0 & B_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & A_{T-1} & -I & 0 & 0 & \cdots & B_{T-1} \\ \hline I & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & I & 0 & 0 & \cdots & 0 \end{array}\right], \quad \tilde{d} = \left[\begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \\ \hline x^{\mathrm{init}} \\ x^{\mathrm{des}} \end{array}\right]$$

Example

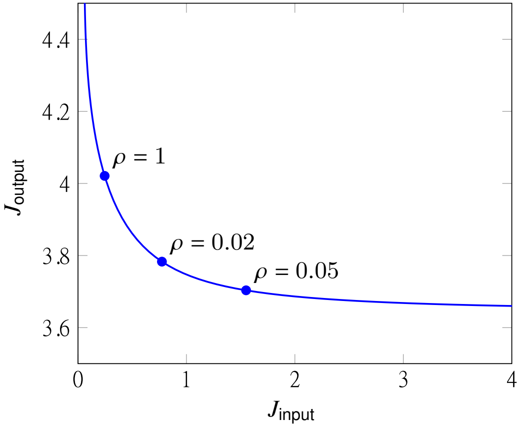
time-invariant system with constant matrices

$$A = \begin{bmatrix} 0.855 & 1.161 & 0.667 \\ 0.015 & 1.073 & 0.053 \\ -0.084 & 0.059 & 1.022 \end{bmatrix}, \quad B = \begin{bmatrix} -0.076 \\ -0.139 \\ 0.342 \end{bmatrix}$$

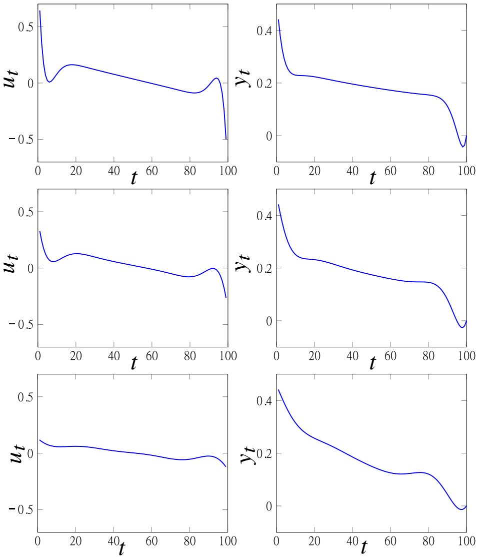
$$C = \begin{bmatrix} 0.218 & -3.597 & -1.683 \end{bmatrix}$$

- $y^{\mathrm{des}} = 0, T = 100$
- initial condition $x^{\mathrm{init}} = (0.496, -0.745, 1.394)$
- target or desired final state $x^{\mathrm{des}} = 0$
- input and output have dimension one

Optimal trade-off curve



Three points on the trade-off curve



Linear state feedback control

Linear state feedback

- *linear state feedback control* uses the input

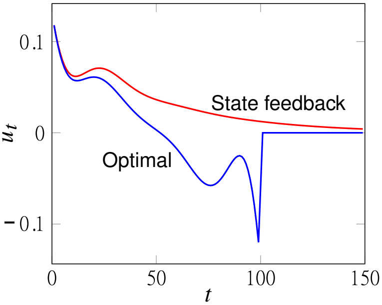
$$u_t = Kx_t, \quad t = 1, 2, \ldots$$

- K is the *state feedback gain matrix*
- widely used, especially when x_t should converge to zero, T is not specified

One approach to compute K

- solve the linear quadratic control problem with $x^{\text{des}} = 0$ for (large) T
- solution u_t is a linear function of x^{init}, hence u_1 can be written as $u_1 = Kx^{\text{init}}$
- columns of K can be found by computing u_1 for $x^{\text{init}} = e_1, \ldots, e_n$
- use this K as state feedback gain matrix

Example



- setup of previous example
- blue curve uses optimal linear quadratic control for $T = 100$
- red curve uses simple linear state feedback $u_t = Kx_t$
- optimal choice achieves $y_T = 0$ but linear state feedback makes y_T small only

Outline

- constrained least squares
- solution of least norm problem
- solution of constrained least squares
- linear quadratic control
- **linear quadratic estimation**
- portfolio optimization

State estimation

Linear dynamical system model

$$x_{t+1} = A_t x_t + B_t w_t, \quad y_t = C_t x_t + v_t, \quad t = 1, 2, \ldots$$

- x_t is state (n-vector)
- y_t is measurement (p-vector)
- w_t is input or process noise (m-vector)
- v_t is measurement noise or residual (p-vector)
- A_t, B_t, C_t are the known dynamics, input, and output matrices

State estimation

- we have measurements $y_1, \ldots, y_T$
- w_t, v_t are unknown, but assumed small
- goal: estimate state sequence $x_1, \ldots, x_T$

Least squares state estimation

$$\begin{array}{ll} \text{minimize} & J_{\text{meas}} + \lambda J_{\text{proc}} \\ \text{subject to} & x_{t+1} = A_t x_t + B_t w_t, \quad t = 1, \dots, T-1 \end{array}$$

- variables are the states $x_1, \dots, x_T$ and input noise $w_1, \dots, w_{T-1}$
- primary objective J_{meas} is sum of squares of measurement residuals:

$$J_{\text{meas}} = \|C_1 x_1 - y_1\|^2 + \cdots + \|C_T x_T - y_T\|^2$$

- secondary objective J_{proc} is sum of squares of process noise

$$J_{\text{proc}} = \|w_1\|^2 + \cdots + \|w_{T-1}\|^2$$

- $\lambda > 0$ is a parameter, trades off measurement and process errors
- similar to control formulation but interpretation is different

Constrained least squares formulation

$$\begin{array}{ll}\text{minimize} & \|C_1x_1 - y_1\|^2 + \cdots + \|C_T x_T - y_T\|^2 + \lambda\left(\|w_1\|^2 + \cdots + \|w_{T-1}\|^2\right) \\ \text{subject to} & x_{t+1} = A_t x_t + B_t w_t, \quad t = 1, \ldots, T-1\end{array}$$

- can be written as

$$\begin{array}{ll}\text{minimize} & \|\tilde{A}z - \tilde{b}\|^2 \\ \text{subject to} & \tilde{C}z = \tilde{d}\end{array}$$

- vector z contains the $Tn + (T-1)m$ variables:

$$z = (x_1, \ldots, x_T, w_1, \ldots, w_{T-1})$$

Constrained least squares formulation

$$\tilde{A} = \left[\begin{array}{cccc|ccc} C_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & C_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & C_T & 0 & \cdots & 0 \\ \hline 0 & 0 & \cdots & 0 & \sqrt{\lambda} I & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & \sqrt{\lambda} I \end{array}\right], \quad \tilde{b} = \left[\begin{array}{c} y_1 \\ y_2 \\ \vdots \\ y_T \\ \hline 0 \\ \vdots \\ 0 \end{array}\right]$$

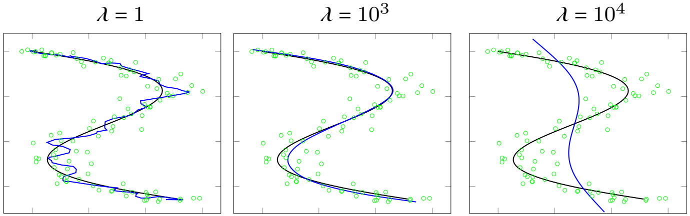
$$\tilde{C} = \left[\begin{array}{cccccc|cccc} A_1 & -I & 0 & \cdots & 0 & 0 & B_1 & 0 & \cdots & 0 \\ 0 & A_2 & -I & \cdots & 0 & 0 & 0 & B_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & A_{T-1} & -I & 0 & 0 & \cdots & B_{T-1} \end{array}\right], \quad \tilde{d} = 0$$

Example

$$A_t = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_t = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

- simple model of mass moving in a 2-D plane
- $x_t = (p_t, z_t)$: 2-vector p_t is position, 2-vector z_t is the velocity
- $y_t = C_t x_t + w_t$ is noisy measurement of mass position
- $T = 100$

Position estimates



- 100 noisy measurements y_t shown as circles
- solid line is exact position $C_t x_t$
- blue lines show position estimates for three values of λ

Outline

- constrained least squares
- solution of least norm problem
- solution of constrained least squares
- linear quadratic control
- linear quadratic estimation
- **portfolio optimization**

Return of an asset

Asset value

- asset can be stock, bond, real estate, commodity, ...
- buy q shares of an asset at price p at beginning of investment period
- $h = pq$ is dollar value of holdings

Asset return

- sell q shares at new price p^+ at end of period
- profit is

$$q(p^+ - p) = \frac{(p^+ - p)}{p} h = rh$$

where r (fractional) return is

$$r = \frac{(p^+ - p)}{p} = \frac{\text{profit}}{\text{investment}}$$

Mean return and risk

- r is a time-series (vector) of returns
- $\mathrm{avg}(r)$ is portfolio *mean return* (or just return); $\mathrm{std}(r)$ is *risk*
- $\mathrm{avg}(r)$ and $\mathrm{std}(r)$ are *per-period* return and risk
- mean return and risk are often expressed in annualized form (*i.e.*, per year)

Annualized return and risk: if we have P trading periods per year

$$\text{annualized return } = P\,\mathrm{avg}(r), \quad \text{annualized risk } = \sqrt{P}\mathrm{std}(r)$$

- if returns are daily, with 250 trading days in a year

$$\text{annualized return } = 250\mathrm{avg}(r), \quad \text{annualized risk } = \sqrt{250}\mathrm{std}(r)$$

- example: daily return r with per-period (daily) return 0.05% and risk 0.5% has an annualized return and risk of 12.5% and 7.9%

Portfolio investment

- n different assets
- we invest a total of V dollars over some period (one day, week, month, ...)
- goal: make investments so that the combined return for all investments is high

Portfolio allocation weights

- w is *asset weight* or *allocation vector* with $\mathbf{1}^T w = 1$
- w_j is fraction of total portfolio value held in asset j; short position if $w_j < 0$
  - short positions are assets you borrow and sell at the beginning, but must return to the borrower at the end of the period
- Vw_j is the dollar value of asset j
- $w = (-0.2, 0.0, 1.2)$ means we take a short position of $0.2V$ in asset 1, don't hold any of asset 2, and invest $1.2V$ in asset 3
- *leverage* of portfolio is $L = |w_1| + \cdots + |w_n|$

Return matrix

(asset) *return matrix* for investments held for T periods is

$$R = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1n} \\ R_{21} & R_{22} & \cdots & R_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ R_{T1} & R_{T2} & \cdots & R_{Tn} \end{bmatrix} = \begin{bmatrix} \tilde{r}_1^T \\ \tilde{r}_2^T \\ \vdots \\ \tilde{r}_T^T \end{bmatrix}$$

- R_{tj} is fractional return of asset j in period t
  - $R_{61} = 0.02$ means that asset 1 gained 2% in period 6
- tth row $\tilde{r}_t^T$ gives asset returns in period t
- jth column is time series of asset j returns
- we often assume asset n is cash with risk-free return $\mu^{\mathrm{rf}} > 0$
- if last asset is risk-free, the last column of R is $\mu^{\mathrm{rf}}\mathbf{1}$

Return over a period

- we invest a total (positive) amount V_t at the beginning of period t
- so we invest $V_t w_j$ in asset j
- the dollar value of the whole portfolio at end of period t is

$$V_{t+1} = \sum_{j=1}^{n} V_t w_j \left(1 + R_{tj}\right) = V_t(1 + \tilde{r}_t^T w)$$

  where $\tilde{r}_t = (R_{t1}, \ldots, R_{tn})$

- total (fractional) return of the portfolio over period t is

$$\frac{V_{t+1} - V_t}{V_t} = \frac{V_t(1 + \tilde{r}_t^T w) - V_t}{V_t} = \tilde{r}_t^T w$$

- $r = Rw$ is portfolio (fractional) returns vector (time series)
  - if n is risk free and $w = e_n$, then $Rw = \mu^{\mathrm{rf}}\mathbf{1}$ (constant return)

Portfolio value

Total portfolio value: if r is portfolio return vector in period t, then

$$V_{t+1} = V_1\,(1 + r_1)\,(1 + r_2) \cdots (1 + r_t)$$

- V_1 is initial investment amount
- portfolio value versus time traditionally plotted using $V_1 = \$10000$

Approximate total portfolio value

- for small per-period returns r_t and not too large T, we have

$$\begin{aligned} V_{T+1} &= V_1\,(1 + r_1) \cdots (1 + r_T) \\ &\approx V_1 + V_1\,(r_1 + \cdots + r_T) \\ &= V_1(1 + T\,\mathrm{avg}(r)) \end{aligned}$$

- approximation assumes $r_i r_j$ are small (*e.g.*, $|r_t|$ small) and can be neglected
- approx. suggests that we can maximize our portfolio value, by maximizing $\mathrm{avg}(r)$

Portfolio optimization

choose w to minimize risk with fixed mean return ρ

$$\begin{array}{ll} \text{minimize} & \mathbf{std}(Rw)^2 = (1/T)\|Rw - \rho\mathbf{1}\|^2 \\ \text{subject to} & \mathbf{1}^T w = 1, \quad \mathbf{avg}(Rw) = \rho \end{array}$$

- R is the returns matrix for past returns
- $r = Rw$ is the (past) portfolio return time series
- solutions w are called *Pareto optimal*

Assumption: *future returns will be similar to past ones*

- this is false in general
- we choose w that would have worked well on past returns
- ... and hope it will work well going forward (just like data fitting)
- we can use validation by finding a solution of certain past period, then testing on another past period

Portfolio optimization via constrained least squares

$$\begin{array}{ll} \text{minimize} & \|Rw - \rho\mathbf{1}\|^2 \\ \text{subject to} & \begin{bmatrix} \mathbf{1}^T \\ \mu^T \end{bmatrix} w = \begin{bmatrix} 1 \\ \rho \end{bmatrix} \end{array}$$

- $\mu = (1/T)R^T\mathbf{1}$ is n-vector of (past) asset returns
- ρ is required (past) portfolio return
- an equality constrained least squares problem, with solution

$$\begin{bmatrix} w \\ z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 2R^TR & \mathbf{1} & \mu \\ \mathbf{1}^T & 0 & 0 \\ \mu^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 2\rho T\mu \\ 1 \\ \rho \end{bmatrix}$$

Optimal portfolio

optimal portfolio w is an affine function of ρ

$$\begin{bmatrix} w \\ z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 2R^TR & \mathbf{1} & \mu \\ \mathbf{1}^T & 0 & 0 \\ \mu^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + \rho \begin{bmatrix} 2R^TR & \mathbf{1} & \mu \\ \mathbf{1}^T & 0 & 0 \\ \mu^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 2T\mu \\ 0 \\ 1 \end{bmatrix}$$

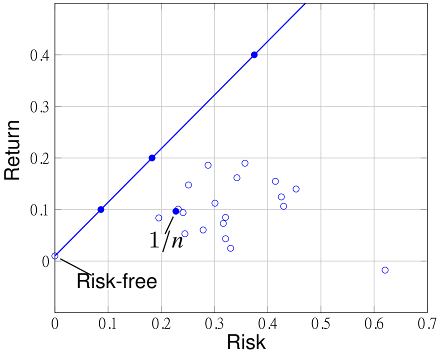
vector w has the form

$$w = w^0 + \rho v, \quad \mathbf{1}^T v = 0$$

- Pareto optimal portfolio form a line with base w^0 and direction v
- a point on a line can be written as affine combination of two other points on line
- Pareto optimal portfolios are affine comb. of just two portfolios (two-fund theorem)

Example

- daily return data for 19 stocks over a period of 2000 days (8 years)
- plus risk-free asset with 1% annual return
- open circles shows individual assets $(\sqrt{250}\text{std}(Re_i), 250\text{avg}(Re_i))$
- line shows risk and return for the Pareto optimal portfolios (for different ρ)

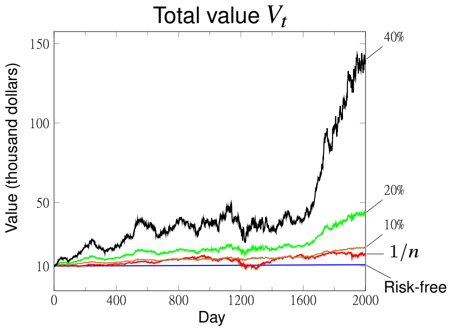


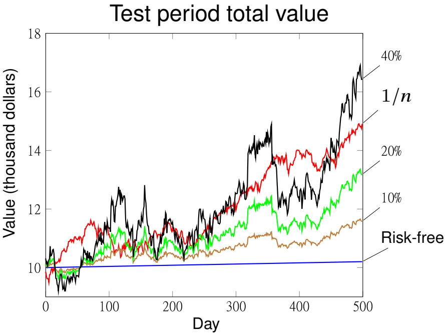
Five portfolios

Portfolio	Return		Risk		Leverage
	Train	Test	Train	Test	
risk-free	0.01	0.01	0.00	0.00	1.00
$\rho = 10\%$	0.10	0.08	0.09	0.07	1.96
$\rho = 20\%$	0.20	0.15	0.18	0.15	3.03
$\rho = 40\%$	0.40	0.30	0.38	0.31	5.48
$1/n$ (uniform weights)	0.10	0.21	0.23	0.13	1.00

- train period of 2000 days used to compute optimal portfolio
- test period is different 500-day period

Total portfolio value





References and further readings

- S. Boyd and L. Vandenberghe. *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares,* Cambridge University Press, 2018.

- L. Vandenberghe. *EE133A lecture notes,* Univ. of California, Los Angeles. (`http://www.seas.ucla.edu/~vandenbe/ee133a.html`)