Least squares in Julia

Reese Pathak   Stephen Boyd

EE103
Stanford University

November 15, 2016

Outline

Least squares

Multi-objective least squares

Linearly constrained least squares

Least squares approximate solution in Julia

the math:

- $\hat{x}$ minimizes $\|Ax - b\|^2$; A has independent columns
- $\hat{x} = (A^T A)^{-1} A^T b = A^\dagger b = R^{-1} Q^T b$
  ($A = QR$ is QR-factorization of A)

in Julia:

- `xhat = inv(A'*A)*(A'*b)`
- `xhat = pinv(A)*b`
- `Q,R = qr(A); xhat = inv(R)*(Q'*b)`
- simplest method: `xhat = A\b`

Example: Regression

- N columns of X are feature n-vectors
- N-vector y gives associated outcomes
- regression model: find n-vector β, scalar v that minimize

$$\|X^T\beta + v\mathbf{1} - y\|^2$$

- express objective as

$$\left\| \begin{bmatrix} \mathbf{1} & X^T \end{bmatrix} \begin{bmatrix} v \\ \beta \end{bmatrix} - y \right\|^2$$

- in Julia:

```
beta_tilde = [ ones(N,1) X' ] \ y;
v = beta_tilde[1];  beta = beta_tilde[2:end];
```

The backslash operator

the backslash operator `x = A\b` is heavily overloaded

- if A is square and invertible
  - $x = A^{-1}b$
  - the unique solution of square set of equations $Ax = b$
- if A is tall with linearly independent columns
  - $x = (A^TA)^{-1}A^Tb$
  - the unique least squares approximate solution of overdetermined equations $Ax = b$
- if A is wide with linearly independent rows
  - $x = A^T(AA^T)^{-1}b$
  - the unique least norm solution of the underdetermined equations $Ax = b$
- in other cases, `A\b` will give an error message

Outline

Least squares

Multi-objective least squares

Linearly constrained least squares

Multi-objective least squares

the math (for two objectives):

- $\hat{x}$ minimizes $\lambda_1\|A_1x - b_1\|^2 + \lambda_2\|A_2x - b_2\|^2$
- $\lambda_1, \lambda_2 > 0$ are relative weights, trade off objectives
- solve by stacking:

$$\hat{x} = \begin{bmatrix} \sqrt{\lambda_1}A_1 \\ \sqrt{\lambda_2}A_2 \end{bmatrix}^\dagger \begin{bmatrix} \sqrt{\lambda_1}b_1 \\ \sqrt{\lambda_2}b_2 \end{bmatrix}$$

- or $\hat{x} = (\lambda_1 A_1^T A_1 + \lambda_2 A_2^T A_2)^{-1}(\lambda_1 A_1^T b_1 + \lambda_2 A_2^T b_2)$

in Julia:

```
sl1=sqrt(lambda1); sl2=sqrt(lambda2);
x_hat = [sl1*A1; sl2*A2) ] \ [sl1*b1; sl2*b2 ]
```

Outline

Least squares

Multi-objective least squares

Linearly constrained least squares

Equality constrained least squares

the math:

- $\hat{x}$ minimizes $\|Ax - b\|^2$ subject to $Cx = d$
- A is $m \times n$, C is $p \times n$
- find $\hat{x}$ by solving $(n+p) \times (n+p)$ KKT system

$$\begin{bmatrix} 2A^T A & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ z \end{bmatrix} = \begin{bmatrix} 2A^T b \\ d \end{bmatrix}$$

in Julia:

```
kkt_sol = [2*A'*A C'; C zeros(p,p)] \ [2*A'*b; d]
x_hat = kkt_sol[1:n]
```

Least norm problem

the math:

- $\hat{x}$ minimizes $\|x\|^2$ subject to $Cx = d$
- can solve by KKT system, or $\hat{x} = C^T(CC^T)^{-1}d = C^\dagger d$

in Julia: `x_hat = C\d`