

Linear Least Squares Modeling in Haskell

Tri Dao

CS 240H
Stanford University

March 16, 2016

Least squares problem

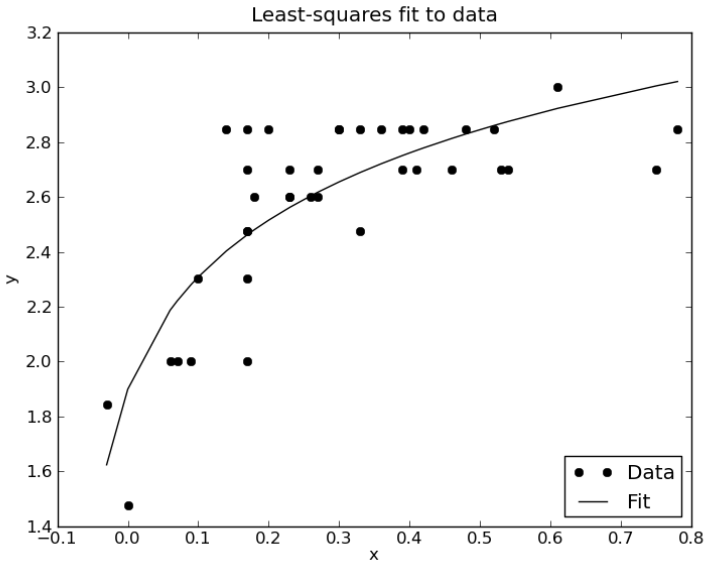
- ▶ The *least squares problem* is

$$\text{minimize } \|Ax - b\|^2$$

- ▶ Solution of least squares problem is

$$\hat{x} = (A^T A)^{-1} A^T b = A^\dagger b$$

Example



Constrained least squares problem

- ▶ The (linearly) *constrained least squares problem* is

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

- ▶ Solution of the constrained least squares problem is

$$\begin{bmatrix} \hat{x} \\ z \end{bmatrix} = \begin{bmatrix} 2A^T A & C^T \\ C & 0 \end{bmatrix}^{-1} \begin{bmatrix} 2A^T b \\ d \end{bmatrix}$$

Least squares modeling

```
import qualified Data.Map.Strict as M
import Numeric.LinearAlgebra
import LeastSquares

minimize :: Expr -> [Expr] -> M.Map VarName (Vector R)
```

Least squares modeling

```
import Data.Map.Strict ((!))
import Numeric.LinearAlgebra hiding ((!))
import LeastSquares

leastSquaresEstimate :: Vector R
leastSquaresEstimate = result ! "x"
  where result = minimize (SumSquares (a*x-b)) [c*x==:d]
        a = Mat $ (3><2) [1.0, 1.0, 1.0, 4.0, 2.0, -1.0]
        b = Vec $ vector [13.0, 27.0, 1.0]
        c = Mat $ (1><2) [3.0, 7.0]
        d = Vec $ vector [-1.0]
        x = Var "x" 2
```

Modeling language

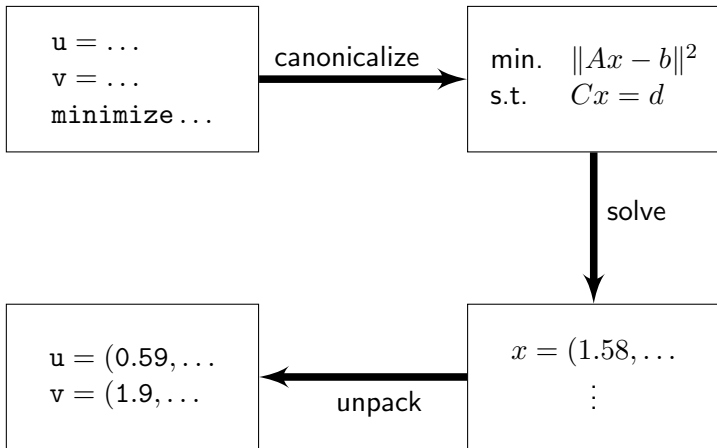


Image de-blurring

- ▶ x is an image, A is a blurring operator, and $y = Ax + v$ is a blurred, noisy image
- ▶ least-squares de-blurring: choose x to minimize

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

D_v , D_h are vertical and horizontal differencing operations

```
minimize (SumSquares (a*x-b)
               + lambda * (SumSquares (dv*x)
                           + SumSquares (dh*x))) []
```

- ▶ λ controls smoothing of de-blurred image

Example

- ▶ left: blurred, noisy image
- ▶ right: regularized inversion with $\lambda = 0.007$

