UiO : **Matematisk institutt**

Det matematisk-naturvitenskapelige fakultet

STK-4051/9051 Computational Statistics Spring 2022
IRLS and ADMM

Instructor: Odd Kolbjørnsen, oddkol@math.uio.no

Last time

- Introduction – (things you will learn)
- Gradient based methods

$$\boxed{\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \boldsymbol{B}\boldsymbol{s}\left(\boldsymbol{\theta}^{(t)}\right)}$$

  – Newton: $\boldsymbol{B} = \boldsymbol{J}\left(\boldsymbol{\theta}^{(t)}\right)^{-1}$
  – Fisher scoring, $\boldsymbol{B} = \boldsymbol{I}\left(\boldsymbol{\theta}^{(t)}\right)^{-1} = \mathrm{E}\left(\boldsymbol{J}\left(\boldsymbol{\theta}^{(t)}\right)\right)^{-1} = \mathrm{Var}\left(\boldsymbol{s}\left(\boldsymbol{\theta}^{(t)}\right)\right)^{-1}$
  – Secant, $\boldsymbol{B}$: discrete approximation of $\boldsymbol{J}\left(\boldsymbol{\theta}^{(t)}\right)^{-1}$
  – BFGS, (Quasi newton, optim in R) $\boldsymbol{B} = -\alpha \boldsymbol{M}$
  – Ascent, $\boldsymbol{B} = \alpha \boldsymbol{I},\ \alpha > 0,$ but small enough
  – Gauss – Newton , linearize around theta, update using linear regression
- Gauss Seidel: Iterate one coordinate at the time
- Other alternatives
  – Fixed point iterations (can also be gradient based) contraction
  – Nelder – Mead (optim in R)
- Know when to stop

GAUSS-NEWTON revisited

Regression: $\min_{\beta} \sum_{i=1}^{n}(y_i - \beta^T x_i)^2$

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x_1} \\ \vdots \\ \boldsymbol{x_n} \end{bmatrix}$$

$$\min_{\beta} (\boldsymbol{y} - \boldsymbol{X}\beta)^T(\boldsymbol{y} - \boldsymbol{X}\beta)$$

$$\frac{\partial}{\partial \beta}: \quad 2 \cdot \boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\beta) = 0$$

$$\beta = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

NL-LS $\min_{\theta} \sum_{i=1}^{n}(y_i - f(z_i, \theta))^2$

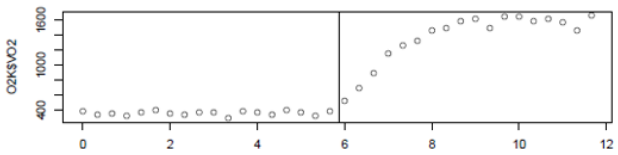
Approx : $\min_{\theta} \sum_{i=1}^{n}(\underbrace{y_i - f(z_i, \theta_k)} - \overbrace{(\theta - \theta_k)}^{}{}^T \underbrace{\nabla f(z_i, \theta_k)})^2$

$$\boldsymbol{f}(\boldsymbol{z}; \boldsymbol{\theta}) = \begin{bmatrix} f(\boldsymbol{z}_1; \boldsymbol{\theta}) \\ \vdots \\ f(\boldsymbol{z}_n; \boldsymbol{\theta}) \end{bmatrix}$$

$$\boldsymbol{A}^{(k)} = \begin{bmatrix} \nabla_\theta f(\boldsymbol{z}_1, \theta_k) \\ \vdots \\ \nabla_\theta f(\boldsymbol{z}_n, \theta_k) \end{bmatrix}$$

$$\theta_{k+1} = \theta_k + \left(\boldsymbol{A}^{(k)^T}\boldsymbol{A}^{(k)}\right)^{-1}\boldsymbol{A}^{(k)^T}(\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{z}; \boldsymbol{\theta}))$$

Solving the problem in R



- All parameters are positive, log-transformed first

```
data(O2K,package="nlstools")
tresh = 5.883
```

```
fit.vo2 = function(param,dat,lambda)
{
  tau1 = exp(param[1])
  beta0 = exp(param[2])
  beta1 = exp(param[3])
  fit = (dat$t <= lambda)*beta0 + (dat$t > lambda) *
    (beta0 + (beta1-beta0)*(1-exp(-(dat$t-lambda) / tau1)))
}
```

Mapping from constrained to unconstrained optimization

```
minusloglik = function(param,dat,tresh=5.883)
{
  fit = fit.vo2(param,dat,tresh)
  sig2 = exp(param[4])
  rss = sum((dat$VO2-fit)^2)
  n = ncol(dat)
  loglik = -0.5*n*log(2*pi)-0.5*n*log(sig2)-0.5*rss/sig2
  -loglik
}
```

Function to be optimized (where first input is the parameters to be optimized)

```
res = optim(c(1,log(400),log(1600),0),minusloglik,dat=O2K,tresh=tresh)
```

Initial guess of parameters to be estimated

Control variables (fixed during optimization)

Discussion

- Why is Newton's method particularly suited for optimization of log likelihoods?
- Why do you think the estimator obtained by **one** iteration of Newton's method is asymptotically just as good as the ML?
- Hint: What is the asymptotic result for ML?

Today

- Iteratively reweighted least square
- ADMM
- ADMM for LASSO

- Combinatorial optimization

Iteratively reweighted least square

- Assume you have data with unknown mean and known variable variance and want to estimate mean

$$\min_{\boldsymbol{\beta}} \sum \frac{\left(y_i - \boldsymbol{x}_i^T \boldsymbol{\beta}_i\right)^2}{\sigma_i^2} = \min_{\boldsymbol{\beta}} \sum w_i \left(y_i - \boldsymbol{x}_i^T \boldsymbol{\beta}_i\right)^2$$

$$\widehat{\boldsymbol{\beta}}_{WLS} = (\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{y} \qquad \boldsymbol{W} = \mathrm{diag}\,(\boldsymbol{w})$$

- Now: what if variance is unknown and a function of the parameter ?

$$\min_{\boldsymbol{\beta}} \sum w_i(\boldsymbol{\beta}, \boldsymbol{x}_i) \left(y_i - \boldsymbol{x}_i^T \boldsymbol{\beta}_i\right)^2$$

- Then: update the weights with previous estimate of parameter.
- Useful trick, can be used for GLM generalized linear models

Weighted least square

$$\min_{\beta} \sum_{i=1}^{n} w_i (y_i - \beta^T x_i)^2$$

$$\min_{\beta} (\boldsymbol{y} - \boldsymbol{X}\beta)^T \boldsymbol{W} (\boldsymbol{y} - \boldsymbol{X}\beta) \qquad \boldsymbol{W} = \begin{bmatrix} w_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & w_n \end{bmatrix}$$

$$\frac{\partial}{\partial \beta}: \quad 2 \cdot \boldsymbol{X}^T \boldsymbol{W} (\boldsymbol{y} - \boldsymbol{X}\beta) = 0$$

$$\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X} \beta = \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{y}$$

$$\beta = (\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{y}$$

Iteratively reweighted least square (linear case)

$$\min_{\boldsymbol{\beta}} \sum w_i(\boldsymbol{\beta}, \boldsymbol{x}_i)\left(y_i - \boldsymbol{x}_i^T \boldsymbol{\beta}_i\right)^2$$

1. $\boldsymbol{w}^{(0)} = \mathbf{1}$
2. For k=1 «until convergence»
   1. $\boldsymbol{\beta}^{(k)} = \operatorname{argmin}\left\{\sum w_i^{(k-1)}\left(y_i - \boldsymbol{x}_i^T \boldsymbol{\beta}_i\right)^2\right\}, \quad i.e.\, \boldsymbol{\beta}^{(k)} = \widehat{\boldsymbol{\beta}}_{WLS}(\boldsymbol{W}^{(k-1)})$
   2. $w_i^{(k)} = w_i\ (\boldsymbol{\beta}^{(k)}, \boldsymbol{x}_i)$

Example: L1-regression robustness to outliers

- Quadratic loss:

$$\arg\min_{\beta} \sum_{i=1}^{n} (y_i - \beta^T x_i\,)^2$$

- Absolute loss:

$$\arg\min_{\beta} \sum_{i=1}^{n} |y_i - \beta^T x_i|$$

$$\sum_{i=1}^{n} |y_i - \beta^T x_i| = \sum_{i=1}^{n} w_i(\beta)(y_i - \beta^T x_i\,)^2 \qquad \boxed{w_i(\beta) = \frac{1}{|y_i - \beta^T x_i|}}$$

If «β is known» we can do weighted least squares regression

* Start with $w_i^{(0)} = 1$. (= least squares regression) to get $\beta^{(0)}$

* In iteration k set $w_i^{(k)} = \frac{1}{|y_i - \beta^{(k-1)T} x_i|}$ or $\min\left\{\frac{1}{|y_i - \beta^{(k-1)T} x_i|}, W_{\max}\right\}$

IRLS.R

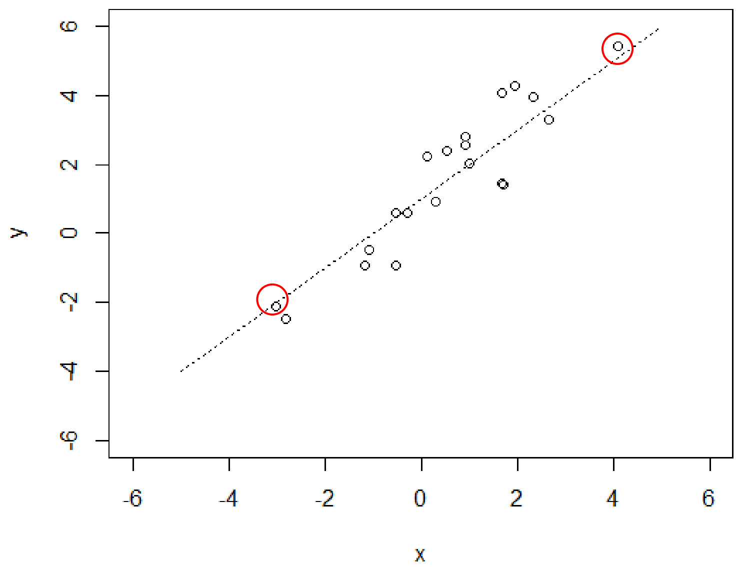
```
IRLS_L1 = function(X,y)
{

  betaHat = solve(t(X)%*%X) %*%(t(X)%*%y )
  pred0 = X%*%betaHat
  res0  =(y-pred)

  betaPrev=c(0,0)
  betaWHat=betaHat
  it=0

  while(sum(abs(betaPrev-betaWHat))>0.0001 & it<100)
  {   maxW=10
      it=it+1
      betaPrev=betaWHat
      pred= Xdata%*%betaPrev
      res=(y-pred)

      w = 1/abs(res)
      w[w>maxW]=maxW  # adjustment to avoid super large numbers [ size relative to problem]
      W = diag(as.vector(w))

      betaWHat = solve(t(Xdata)%*%W%*%Xdata) %*%(t(Xdata)%*%W%*%y )
      #show(betaWHat)
  }
  betaWHat
}
```

$$w_i^{(k)} = \min\left\{\frac{1}{|y_i - \beta^{(k-1)T}x_i|}, W_{\max}\right\}$$

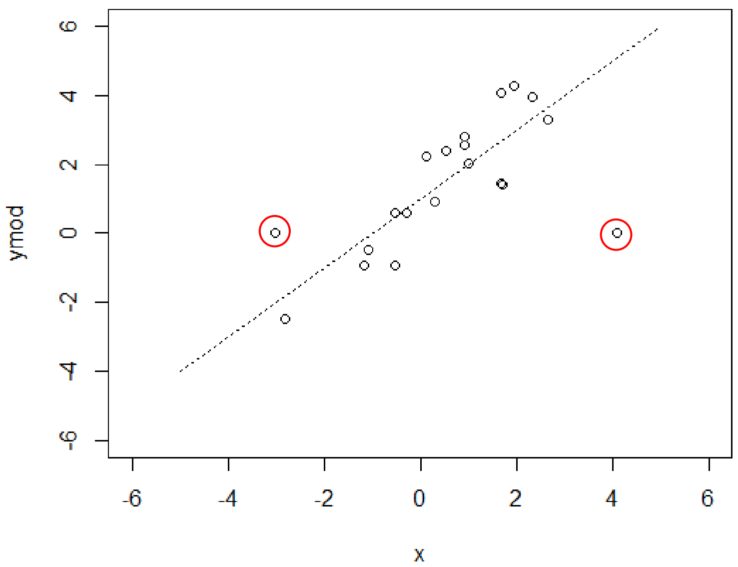
$$\beta = (\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{y}$$

Example n=20

Case 1



Case 2



Example n=20

Quadratic loss

Absolute loss

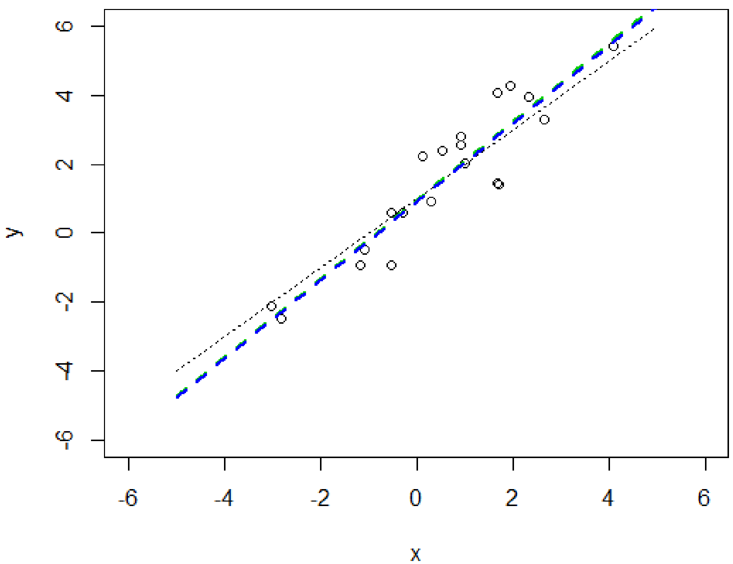
```
> betaHat_L2
       [,1]
  0.9504518
x 1.1409776
```

```
> betaHat_L1
       [,1]
  0.8972019
x 1.1343827
```

```
> betaHat_L2m
       [,1]
  1.0240550
x 0.6887221
```

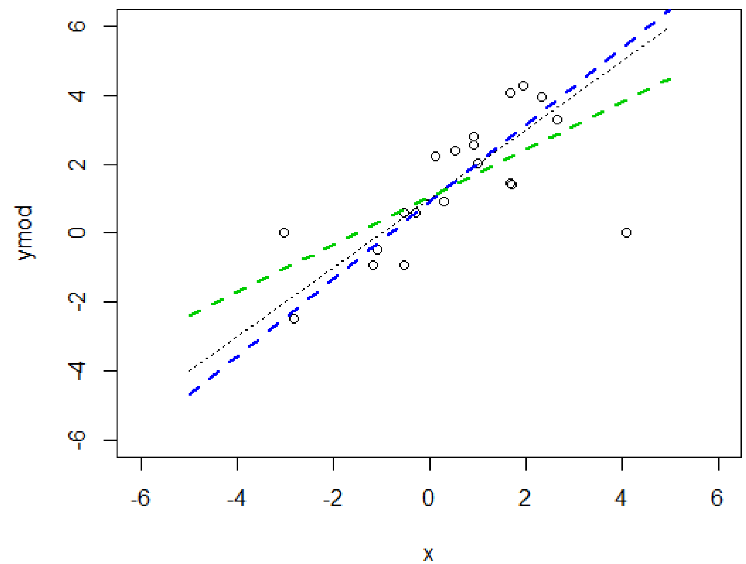
```
> betaHat_L1m
       [,1]
  0.9009297
x 1.1211309
```

Case 1



Results almost identical

Case 2



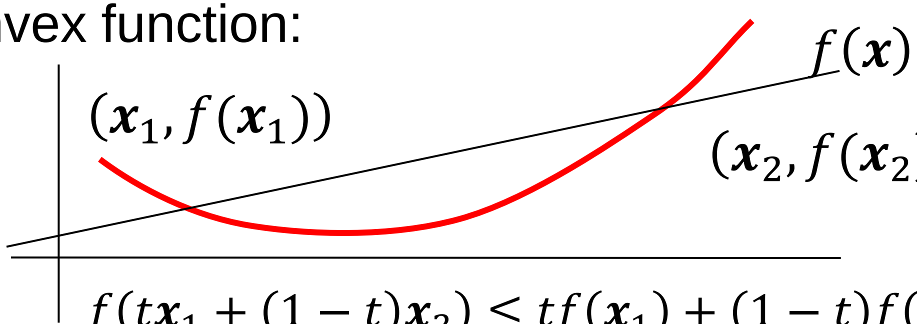
Quadratic loss is sensitive to outliers

Convex optimization

- $$\begin{aligned} &\text{minimize}_{\boldsymbol{x}} && \{ f(\boldsymbol{x}) \} \\ &\quad\text{subject to} && \boldsymbol{a}_i^T \boldsymbol{x} = b_i \\ &\quad\text{and} && g_j(\boldsymbol{x}) \leq 0 \end{aligned}$$
- $i = 1, \ldots, n_a$
- $j = 1, \ldots, n_c$
- $f(\boldsymbol{x}), g_j(\boldsymbol{x})$ convex

https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf

Convex function:



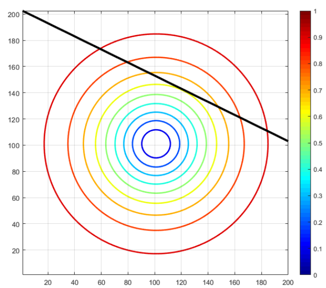
$$f(t\boldsymbol{x}_1 + (1-t)\boldsymbol{x}_2) \leq t f(\boldsymbol{x}_1) + (1-t) f(\boldsymbol{x}_2)$$
$$\text{for } 0 \leq t \leq 1$$

Interpolation always above function

- A field on its own,…
- Covers many important problems
  - Least squares
  - Linear programming
  - Convex quadratic minimization with linear or convex quadratic constraints
  - Conic optimization
  - Second order cone programming
  - …
- Many algorithms open source
  - Bundle methods
  - Subgradient projection methods (Polyak),
  - Interior-point methods
  - Cutting-plane methods
  - Ellipsoid method
  - Subgradient method
  - Dual subgradients
  - …

Just a small taste of convex optimization

Minimization under an equality constraint and the augmented Lagrangian multiplier

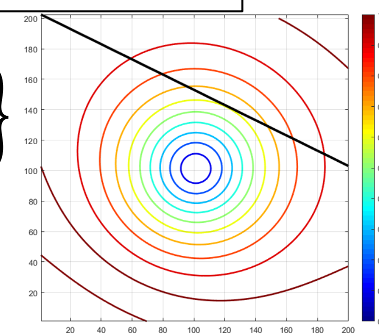


$$\begin{aligned} &\text{minimize}_{\boldsymbol{x}} \quad \{\ f(\boldsymbol{x})\ \} \\ &\text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \end{aligned}$$

$\boldsymbol{x}: (\ p \times 1)$
$\boldsymbol{b}: (\ q \times 1)$
$\boldsymbol{A}: (\ q \times p)$

$$\Updownarrow$$

$$\begin{aligned} &\text{minimize}_{\boldsymbol{x}} \quad \left\{\ f(\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2 \right\} \\ &\text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \end{aligned}$$



$\rho: 1 \times 1$
is a scalar

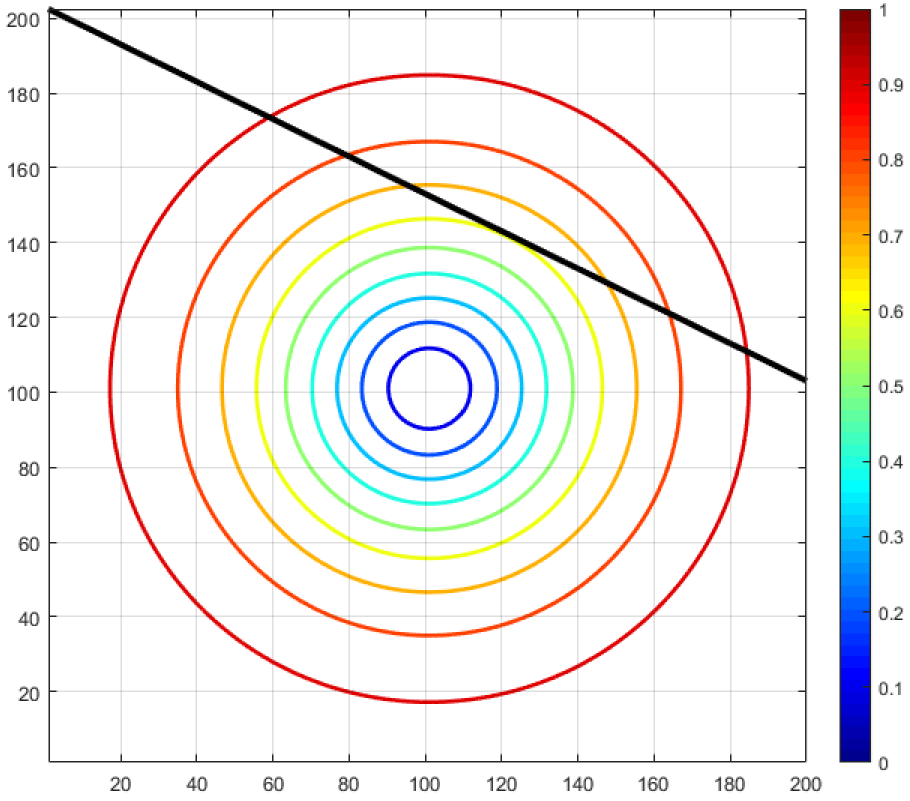
$$\Updownarrow$$

$$\text{minimize}_{\boldsymbol{x},\boldsymbol{\lambda}} \quad \left\{\ f(\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2 + \boldsymbol{\lambda}^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})\right\}$$

$\boldsymbol{\lambda}: (\ q \times 1)$ is the Lagrangian multiplier

Geometric interpretation: Minimum occurs when the contour line is tangent to constraint

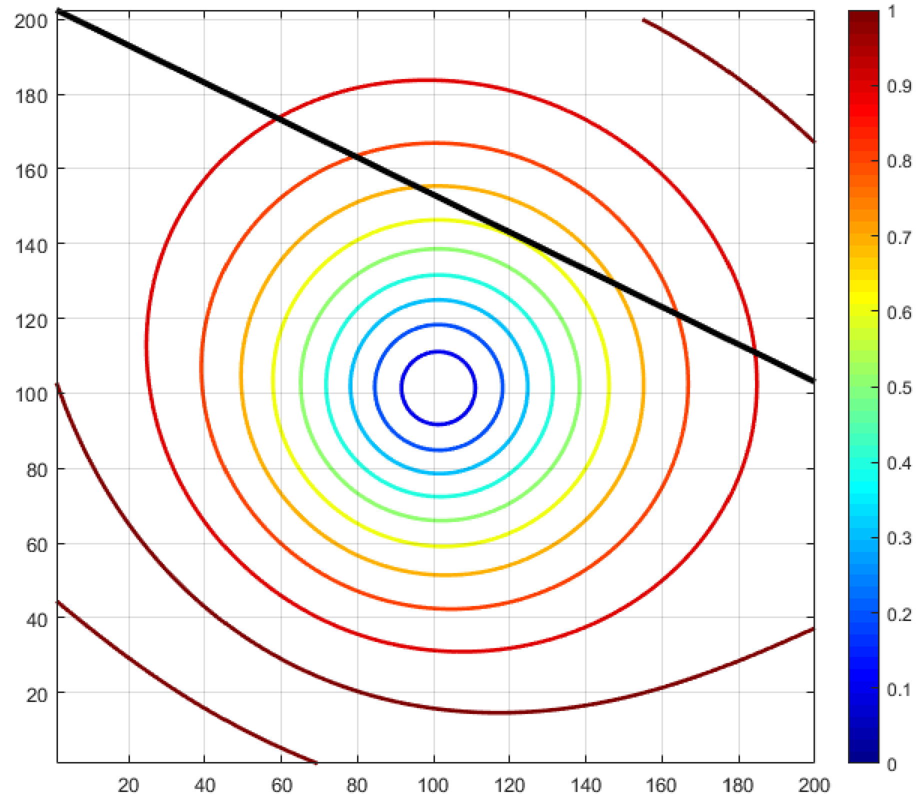
$$\begin{array}{ll} \text{minimize} & \{\, f(\boldsymbol{x}) \,\} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \end{array}$$



$$\text{minimize}_{\boldsymbol{x},\boldsymbol{\lambda}} \quad \left\{ f(\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{A}x - \boldsymbol{b}\|^2 \right\}$$

$$\text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

Does not change along the black line



Algorithm for solution: Method of multipliers

$$\begin{array}{ll} \text{minimize}_{\boldsymbol{x},\boldsymbol{\lambda}} & \left\{ f(\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{A}x - \boldsymbol{b}\|^2 \right\} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \end{array}$$

1. $\boldsymbol{\lambda}^{(0)} = \mathbf{0}$
2. For i=1 «until convergence»
   1. $\boldsymbol{x}^{(i)} = \text{argmin}\left\{f(\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2 + \boldsymbol{\lambda}^{(i-1)T}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})\right\}$
   2. $\boldsymbol{\lambda}^{(i)} = \boldsymbol{\lambda}^{(i-1)} + \rho\left(\boldsymbol{A}\boldsymbol{x}^{(i)} - \boldsymbol{b}\right)$

A version of **dual ascent**, details given in reference for the clever and mathematically inclined student
(part of the article is syllabus for STK9051)

ADMM Alternating Direction Method of Multipliers

$$\text{minimize} \quad \{ f(\boldsymbol{x}) + g(\boldsymbol{z}) \}$$
$$\text{subject to} \quad \boldsymbol{Ax} + \boldsymbol{Bz} = \boldsymbol{c}$$

$$\text{minimize} \quad \left\{ f(\boldsymbol{x}) + g(\boldsymbol{z}) + \boldsymbol{\lambda}^T(\boldsymbol{Ax} + \boldsymbol{Bz} - \boldsymbol{c}\,) + \frac{\rho}{2}\|\boldsymbol{Ax} + \boldsymbol{Bz} - \boldsymbol{c}\|^2 \right\}$$

1. $\boldsymbol{\lambda}^{(0)} = \mathbf{0}, \boldsymbol{z}^{(0)} = \mathbf{0}$
2. For i=1 «until convergence»
   1. $\boldsymbol{x}^{(i)} = \text{argmin}\left\{ f(\boldsymbol{x}) + \frac{\rho}{2}\left\|\boldsymbol{Ax} + \boldsymbol{Bz}^{(i-1)} - \boldsymbol{c}\right\|^2 + \boldsymbol{\lambda}^{(i-1)T}\left(\boldsymbol{Ax} + \boldsymbol{Bz}^{(i-1)} - \boldsymbol{c}\right)\right\}$
   2. $\boldsymbol{z}^{(i)} = \text{argmin}\left\{ g(\boldsymbol{z}) + \frac{\rho}{2}\left\|\boldsymbol{Ax}^{(i)} + \boldsymbol{Bz} - \boldsymbol{c}\right\|^2 + \boldsymbol{\lambda}^{(i-1)T}(\boldsymbol{Ax} + \boldsymbol{Bz} - \boldsymbol{c})\right\}$
   3. $\boldsymbol{\lambda}^{(i)} = \boldsymbol{\lambda}^{(i-1)} + \rho\left(\boldsymbol{Ax}^{(i)} + \boldsymbol{Bz}^{(i)} - \boldsymbol{c}\right)$

Alternate between solving $\boldsymbol{x}$ solving $\boldsymbol{z}$ and update $\boldsymbol{\lambda}$

We do not need to deal with $f(\boldsymbol{x})$ and $g(\boldsymbol{z})$ simultaneously!

- Proven convergence under general assumptions

- Ideal for splitting problem into smaller «solvable» problems

- Syllabus ADMM
  - No proofs
  - Operational use
  - Questions about related stuff

Reference

Foundations and Trends® in
Machine Learning
Vol. 3, No. 1 (2010) 1–122

DOI: 10.1561/2200000016

the essence of knowledge

Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers

Stephen Boyd[1], Neal Parikh[2], Eric Chu[3]
Borja Peleato[4] and Jonathan Eckstein[5]

Available online
https://web.stanford.edu/~boyd/papers/admm_distr_stats.html

Solving lasso using ADMM

The factor $\frac{1}{2}$ is used to simplify expressions

- Lasso:

$$\text{minimize} \quad \left\{\frac{1}{2}\|\boldsymbol{X\beta}-\boldsymbol{y}\|^2+\gamma\|\boldsymbol{\beta}\|_1\right\}$$

$$\Updownarrow$$

$$\text{minimize} \quad \left\{\frac{1}{2}\|\boldsymbol{X\beta}-\boldsymbol{y}\|^2+\gamma\|\boldsymbol{z}\|_1\right\}$$

$$\text{subject to} \quad \boldsymbol{\beta}-\boldsymbol{z}=\boldsymbol{0}$$

In ADMM

1. $$\boldsymbol{\beta}^{(i)}=\operatorname{argmin}\left\{\frac{1}{2}\|\boldsymbol{X\beta}-\boldsymbol{y}\|^2+\frac{\rho}{2}\left\|\boldsymbol{\beta}-\boldsymbol{z}^{(i-1)}\right\|^2+\boldsymbol{\lambda}^{(i-1)T}\left(\boldsymbol{\beta}-\boldsymbol{z}^{(i-1)}\right)\right\}$$

   First problem has only quadratic and linear terms, thus it is solved by linear equations

2. $$\boldsymbol{z}^{(i)}=\operatorname{argmin}\left\{\gamma\|\boldsymbol{z}\|_1+\frac{\rho}{2}\left\|\boldsymbol{\beta}^{(i)}-\boldsymbol{z}\right\|^2+\boldsymbol{\lambda}^{(i-1)T}\left(\boldsymbol{\beta}^{(i)}-\boldsymbol{z}\right)\right\}$$

   Second problem is non linear but can be solved for each component separately

First problem

$$\text{argmin}_\beta \left\{ \frac{1}{2}\|\boldsymbol{X\beta} - \boldsymbol{y}\|^2 + \frac{\rho}{2}\left\|\boldsymbol{\beta} - \boldsymbol{z}^{(i-1)}\right\|^2 + \boldsymbol{\lambda}^{(i-1)T}\left(\boldsymbol{\beta} - \boldsymbol{z}^{(i-1)}\right) \right\}$$

$$(\boldsymbol{X\beta} - \boldsymbol{y})^T(\boldsymbol{X\beta} - \boldsymbol{y}) + \frac{\rho}{2}\left(\boldsymbol{\beta} - \boldsymbol{z}^{(i-1)}\right)^T\left(\boldsymbol{\beta} - \boldsymbol{z}^{(i-1)}\right) + \boldsymbol{\lambda}^{(i-1)T}\left(\boldsymbol{\beta} - \boldsymbol{z}^{(i-1)}\right)$$

$$\frac{\partial}{\partial \boldsymbol{\beta}}: \qquad \boldsymbol{X}^T(\boldsymbol{X\beta} - \boldsymbol{y}) + \rho\left(\boldsymbol{\beta} - \boldsymbol{z}^{(i-1)}\right) + \boldsymbol{\lambda}^{(i-1)} = 0$$

$$(\boldsymbol{X}^T\boldsymbol{X} + \rho\boldsymbol{I})\boldsymbol{\beta} = \boldsymbol{X}^T\boldsymbol{y} + \rho\left(\boldsymbol{z}^{(i-1)} - \frac{\boldsymbol{\lambda}^{(i-1)}}{\rho}\right)$$

$$\boxed{\boldsymbol{\beta} = (\boldsymbol{X}^T\boldsymbol{X} + \rho\boldsymbol{I})^{-\mathbf{1}}\left(\boldsymbol{X}^T\boldsymbol{y} + \rho\left(\boldsymbol{z}^{(i-1)} - \frac{\boldsymbol{\lambda}^{(i-1)}}{\rho}\right)\right)}$$

Second problem

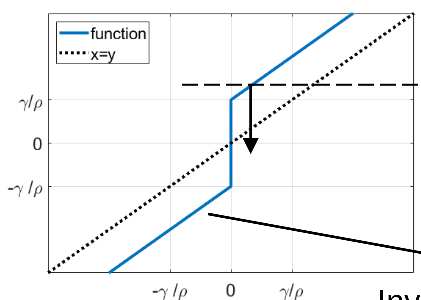
$$\text{argmin}_{\mathbf{z}}\left\{ \gamma\|\boldsymbol{z}\|_1 + \frac{\rho}{2}\left\|\boldsymbol{\beta}^{(i)} - \boldsymbol{z}\right\|^2 + \boldsymbol{\lambda}^{(i-1)T}\left(\boldsymbol{\beta}^{(i)} - \boldsymbol{z}\right)\right\}$$

$$\sum_{j=1}^{p}\left(\gamma|z_j| + \frac{\rho}{2}\left(\beta_j^{(i)} - z_j\right)^2 + \lambda_j^{(i-1)}\left(\beta_j^{(i)} - z_j\right)\right)$$

$$\frac{\partial}{\partial z_j}: \quad \gamma\,\text{sign}(z_j) - \rho\left(\beta_j^{(i)} - z_j\right) - \lambda_j^{(i-1)} = 0$$

$$\left(\frac{\gamma}{\rho}\cdot\text{sign}(z_j) + z_j\right) = \beta_j^{(i)} + \lambda_j^{(i-1)}/\rho$$

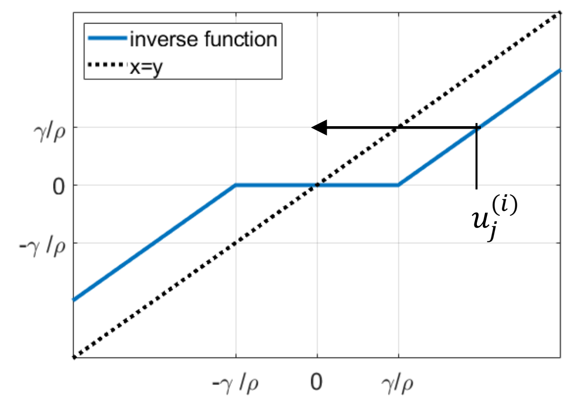
Solve for each component independently



$u_j^{(i)} = \beta_j^{(i)} + \lambda_j^{(i-1)}/\rho$

$$\boxed{z_j^{(i)} = S(u_j^{(i)}; \gamma/\rho)}$$

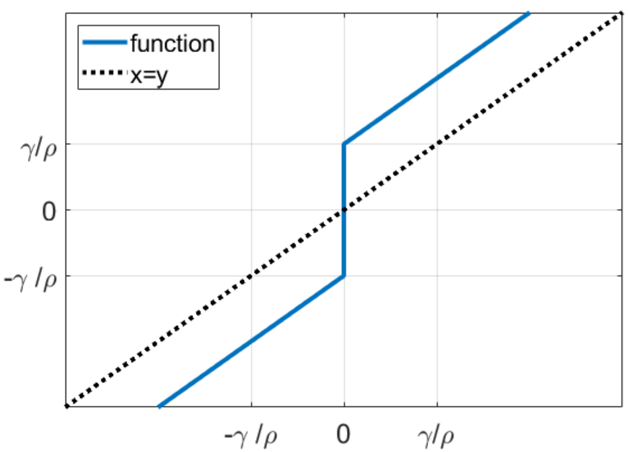
Inverse function

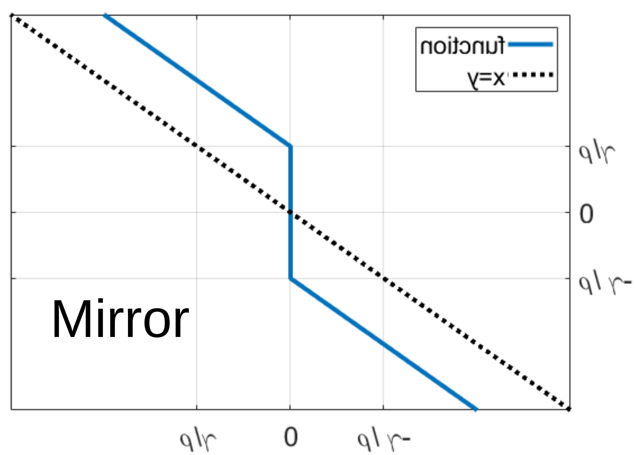


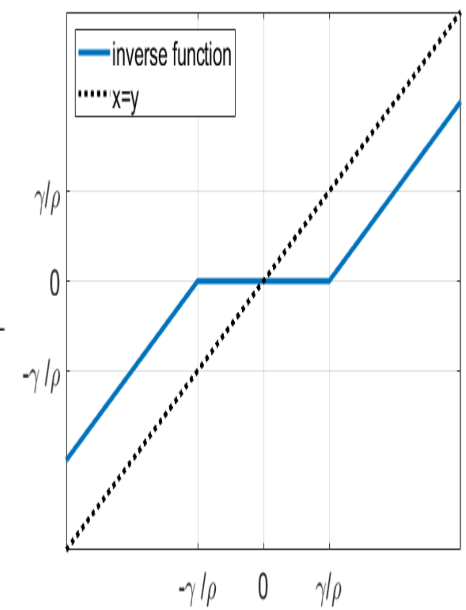
$u_j^{(i)}$

$$S(u;\gamma/\rho) = \text{sign}(u)\cdot\max(|u| - \gamma/\rho, 0)$$

Inverting a function in powerpoint







$$\frac{\gamma}{\rho} \cdot \text{sign}(z) + z$$

$$\text{sign}(u) \cdot \max(|u| - \gamma/\rho, 0)$$

Baseball example R