Constrained Optimization

BTRY/STSCI 4520

Constrained Optimization

In many problems, there are natural constraints on optimization

- Probabilities have $0 \leq p \leq 1$
- Variances $\sigma^2 > 0$

We may be interested in this constraint as a null hypothesis:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \epsilon_i$$

$$H_0 : \beta_1 \leq 1$$

or maybe $H_0 : \beta_1 - \beta_2 \leq 0$.

Also used for model selection:

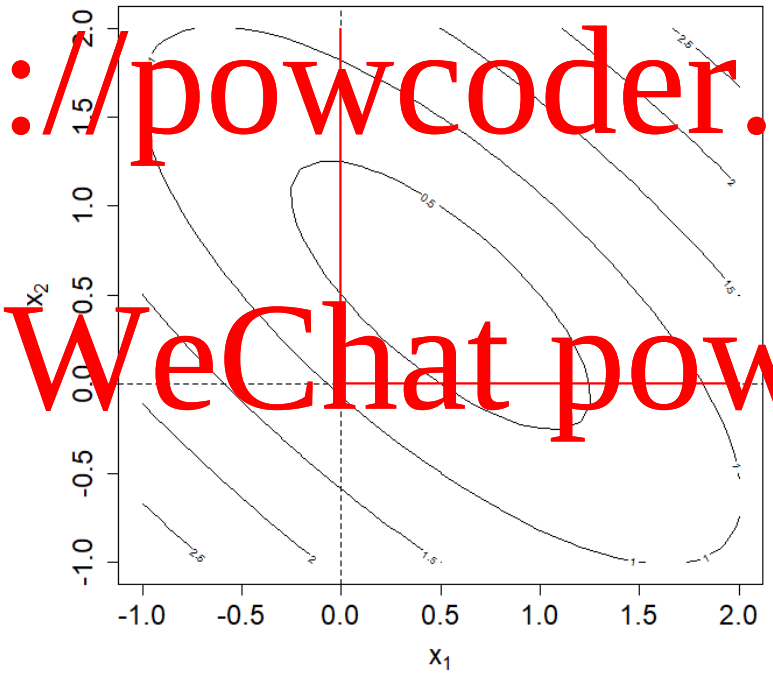
$$\sum |\beta_j| \leq C.$$

But enforcing these constraints can be difficult.

Visual Example

Common problem:

$$\text{minimize } F(x_1, x_2)$$
$$\text{subject to } x_1 \geq 0, \quad x_2 \geq 0$$



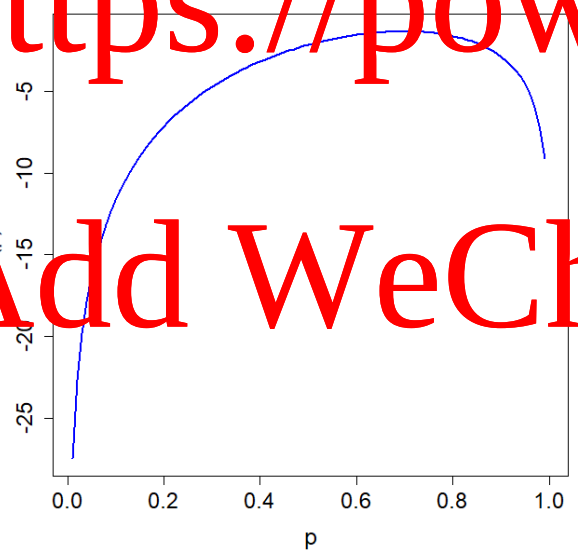
Only the positive quadrant is of interest.

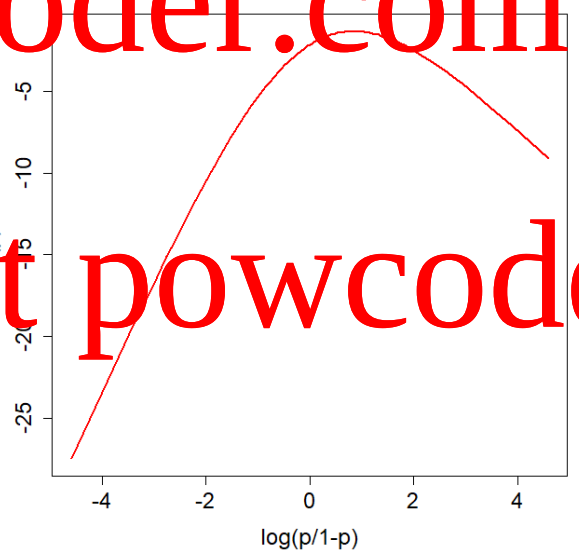
Parameter Transforms

When you expect a minimum inside the constraints: re-represent parameters.

In probabilities

$$p = \frac{e^\theta}{1+e^\theta} \in [0,1] \qquad \theta = \log(\frac{p}{1-p}) \in [-\infty, \infty]$$



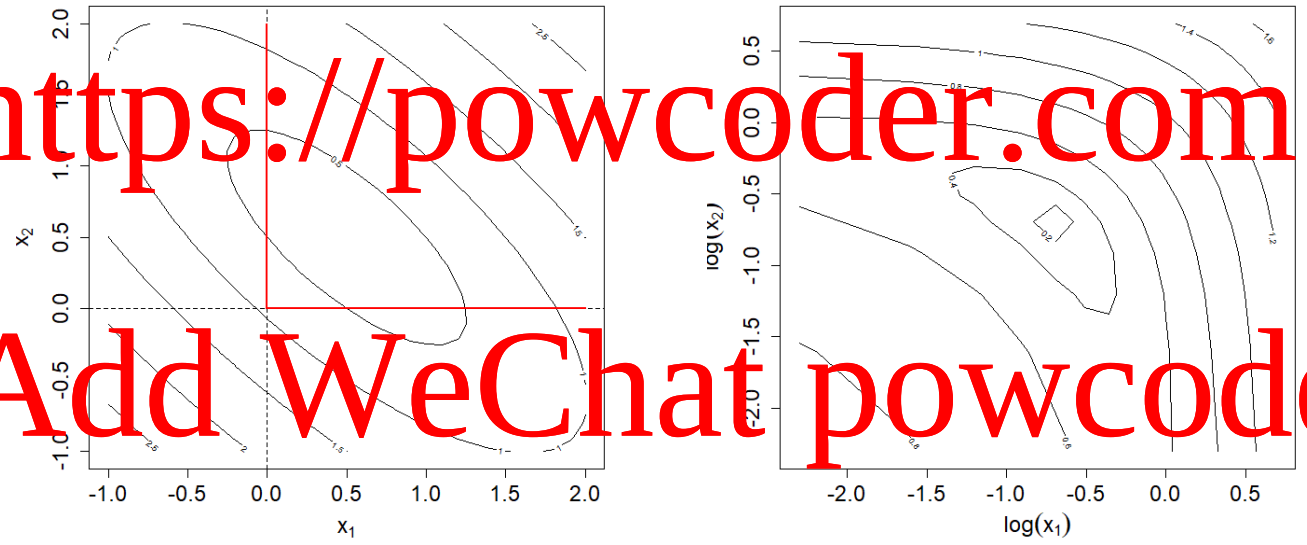


But, may change optimization curvature.

Positive Constraints

Log transformation is common

$$(z_1, z_2) = (\log(x_1), \log(x_2)), \quad \text{minimize } F(e^{z_1}, e^{z_2})$$

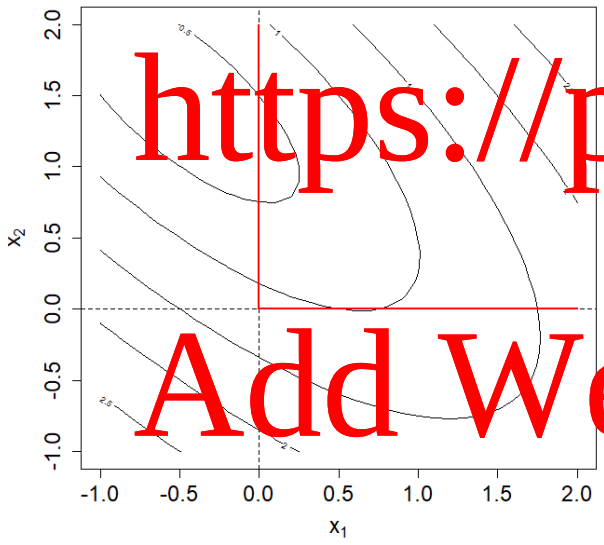


In statistics $\sigma > 0 \to \eta = \log(\sigma) \in [-\infty, \infty]$.

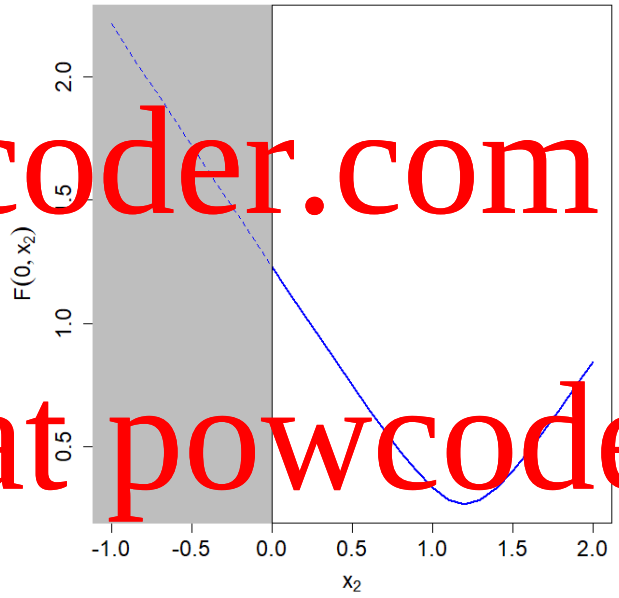
Similar for exponential rates, Gamma, Beta parameters.

What If Constraints are Active?

Sometimes, optimum lies over the boundary:



So the constrained optimum is the minimum on the boundary:



May need to be able to hit the boundary exactly.

When Constraints (and Optimizer) are Nice

Some methods allow linear boundaries, so you can require

$$A\mathbf{x} \geq 0$$

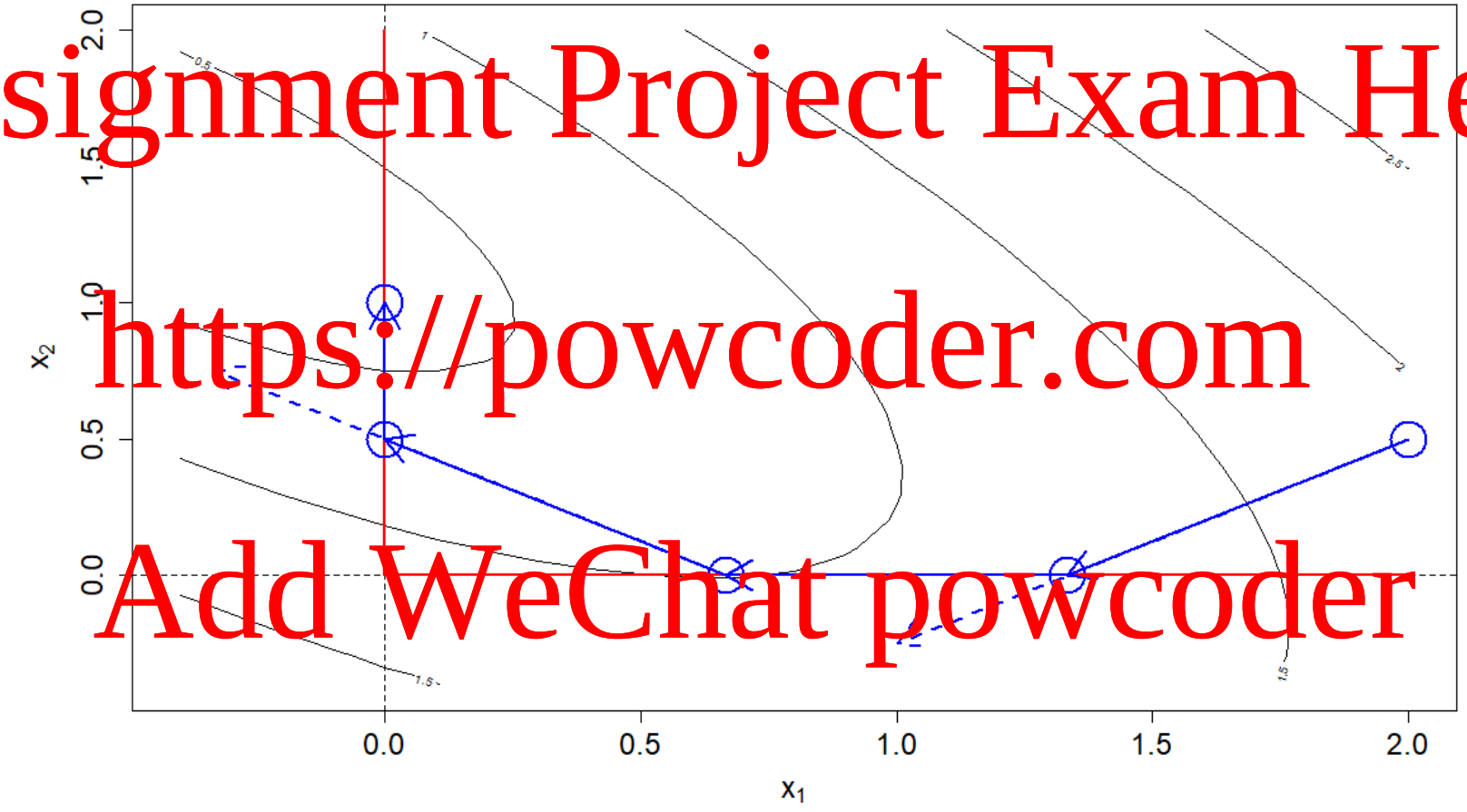
(in our case $A = I$) when optimizing for $\mathbf{x}$.

Separate out interior versus boundary starting points.

- If at the interior
  - Take a proposed optimization step (say, Newton-Raphson)
  - If you cross the boundary, back-track to it.
- If on the boundary
  - Calculate an optimization step.
  - If step is into interior, keep it.
  - Otherwise step along the boundary.

Lots of variations possible (eg check that back-tracking still improves your objective function).

Graphically



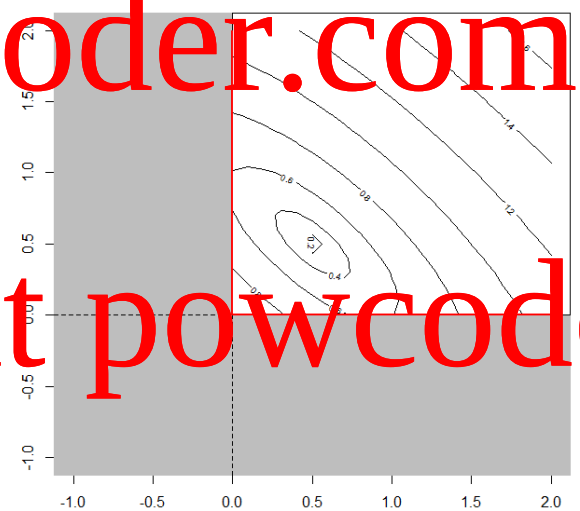
(Steps do not correspond to specific optimization algorithm).

Modified Objective Functions

Make F infinite (or very large) outside constraints:

$$F(x_1, x_2) = F(x_1, x_2) + \infty 1_{x_1<0} + \infty 1_{x_2<0}$$

- Works for simulated annealing or Nelder-Mead.
- Derivatives/secant methods
- Generally won't put you exactly on boundary.



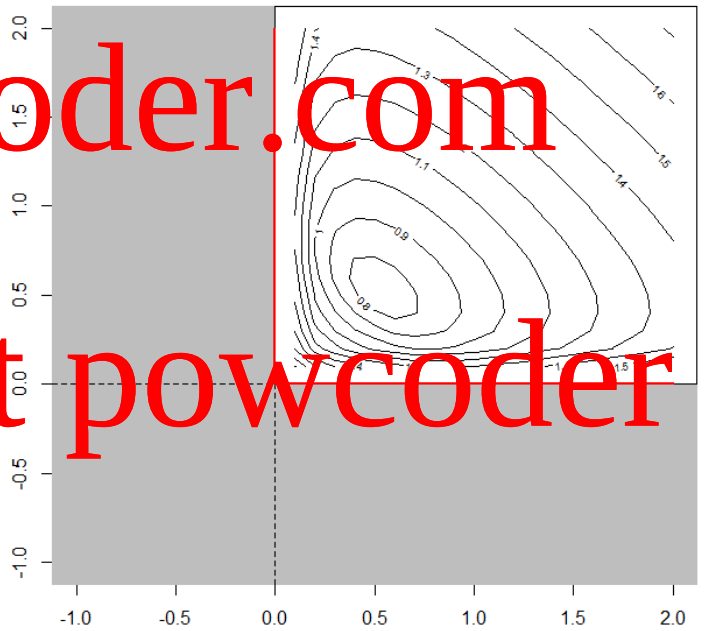
A Sequence of Boundaries

Can make boundaries softer with

$$\tilde{F}(x_1, x_2) = F(x_1, x_2) + 1/(kx_1) + 1/(kx_2)$$

- Solve a sequence of problems with increasing k.
- Optimum converges as $k \to \infty$.
- Can also be used for additional nonlinear constraints:

$$\begin{aligned} &\text{minimize } F(\mathbf{x}) \\ &\text{subject to } G(\mathbf{x}) \geq 0 \\ &\text{and } H(\mathbf{x}) = 0 \end{aligned}$$



In Model Selection

In linear regression

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \epsilon_i, \quad i = 1, \ldots, n$$

when p is large (possibly $p > n$) we may want to set some $\beta_j = 0$.

Recent solution either constrain:

$$\text{minimize} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

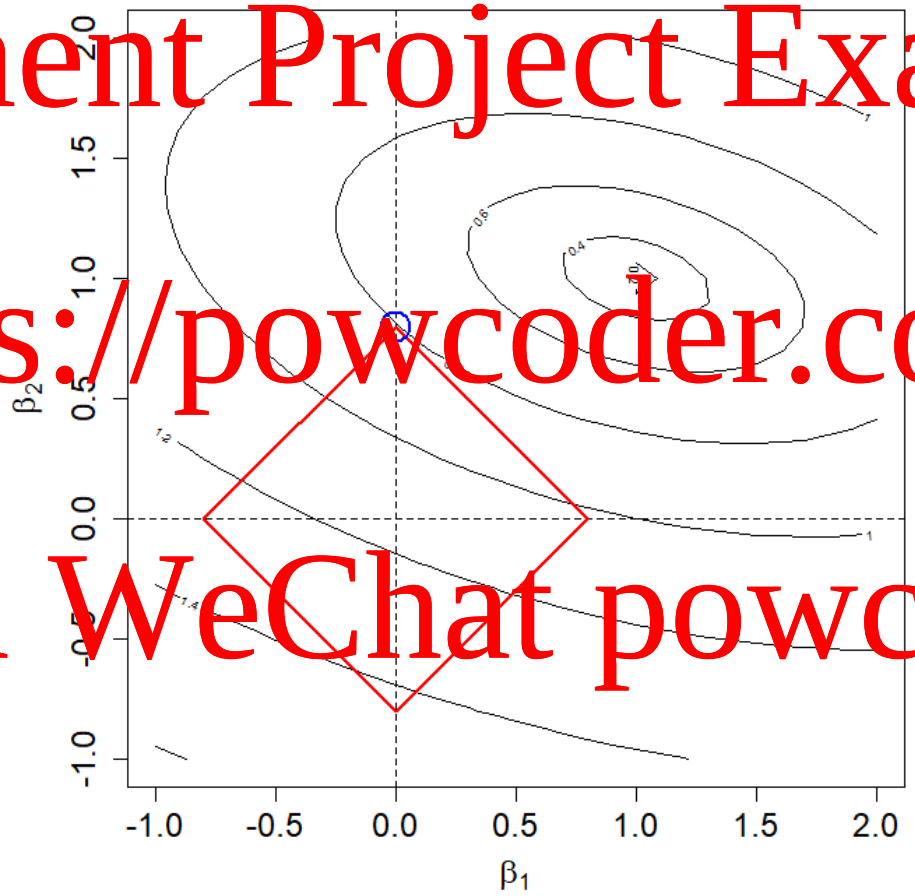
$$\text{subject to} \sum_{j=1}^{p} |\beta_j| < C$$

or penalize (equivalent)

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Why The LASSO?

Least Absolute Subset Selection Operator (Tibhsirani 1996)



"Corners" in $\sum |\beta_j|$ tend to set coefficients exactly to zero.

Obtaining Estimates

Recent computing focussed on penalized form:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

Simplification with 1 covariatiate

$$\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2 + \lambda|\beta_1|$$

Center y_i, x_i; then $\beta_0 = \bar{y} - \beta_1\bar{x} = 0$ gives

$$\sum_{i=1}^{n}(y_i - \beta_1 x_i)^2 + \lambda|\beta_1|$$

Also scale x_i so that $\sum x_i^2 = 1$.

Look at a minimum in 1 dimension.

Non-differentiable Minima

We know that $g(\beta_1) = |\beta_1|$ has a minimum at $\beta_1 = 0$.

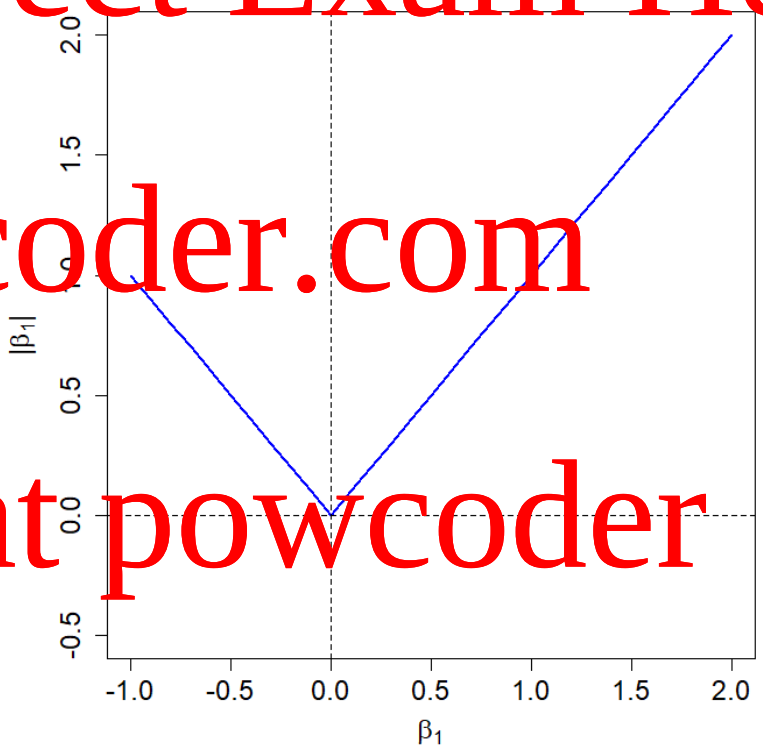
How? It isn't differentiable at 0.

$$\frac{d}{d\beta_1}|\beta_1| = \begin{cases} 1 & \text{if } \beta_1 > 0 \\ -1 & \text{if } \beta_1 < 0 \end{cases}$$

Derivative change sign at 0.

Decreasing as β_1 approaches zero from left, increasing as it leaves to right.

True arbitrarily close to 0.

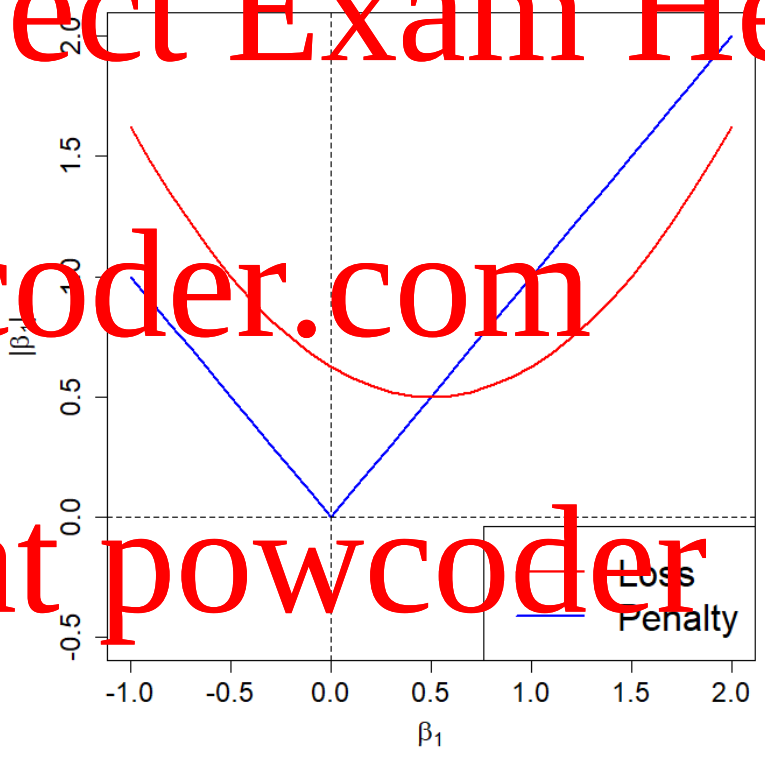


Combining Loss and Penalty

Objective is a combination of

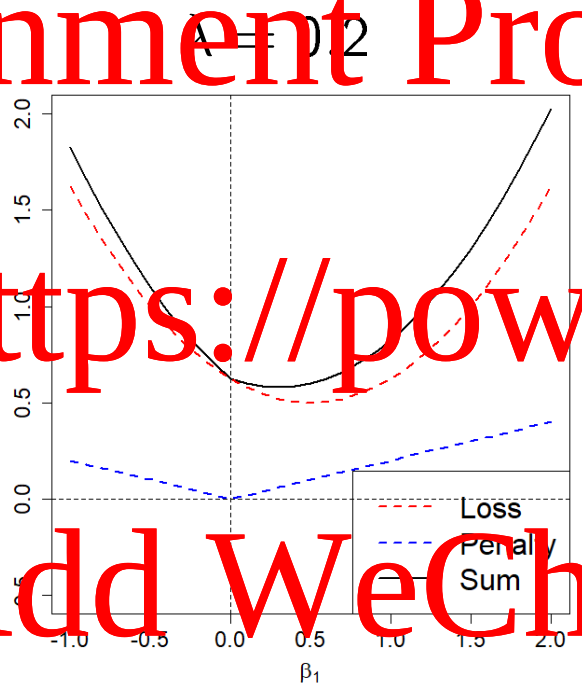
Loss $\sum_i (y_i - x_i\beta_1)^2$

Penalty $\lambda|\beta_1|$

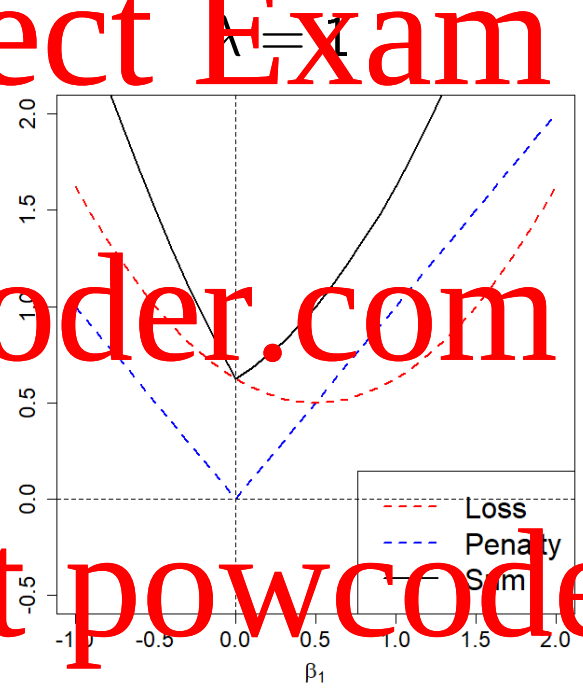


Depending on λ, penalty may keep β_1 at 0 or not.

Illustration



Minimum outside 0



Minimum at 0

Derivatives

$$\frac{d}{d\beta_1}\left[\sum_{i=1}^{n}(y_i - x_i\beta_1)^2 + \lambda|\beta_1|\right] = \begin{cases} -\sum 2x_i(y_i - x_i\beta_1) + \lambda & \text{if } \beta_1 > 0 \\ -\sum 2x_i(y_i - x_i\beta_1) - \lambda & \text{if } \beta_1 < 0 \end{cases}$$

Changes sign at 0 if

$$\left|\sum 2x_i y_i\right| < \lambda$$

otherwise the minimum is at

$$\hat{\beta}_j = \begin{cases} \frac{\sum x_i y_i}{\sum x_i^2} - \frac{\lambda}{2} & \text{if } \sum x_i y_i > 0 \\ \frac{\sum x_i y_i}{\sum x_i^2} + \frac{\lambda}{2} & \text{if } \sum x_i y_i < 0 \end{cases} = \sum x_i y_i - \frac{\lambda}{2}\text{sgn}(\sum x_i y_i)$$

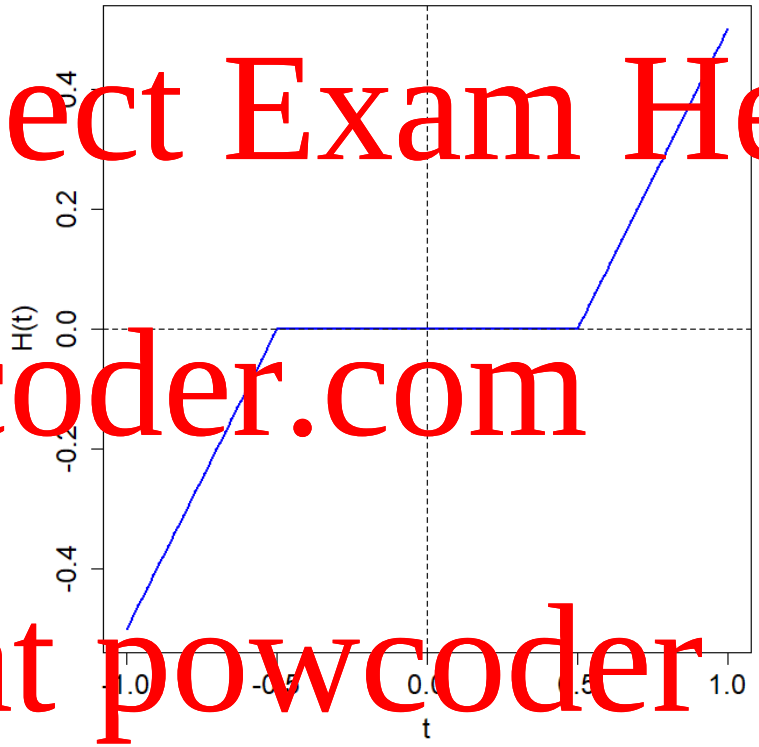
when we have $\sum x_i^2 = 1$.

Soft Thresholding

Often write $\hat{\beta}_j = H_\lambda(\sum x_i y_i)$ where H_λ is the soft thresholding function

$$H_\lambda(t) = t - \lambda \text{sgn}(t)$$

(note we drop the $\lambda/2$ – just redefine λ)




```
ST = function(t,lambda){
        return( max(min(t+lambda,0),t-lambda) )
}
```

A Co-ordinate Descent Strategy

Returning to multiple covariates, our objective is

$$\sum \left(y_i - \sum x_{ij}\beta_j \right)^2 + \lambda \sum |\beta_j|$$

y's, x's centered, scaled.

Written for one β_k, this is

$$\sum \left(y_i - \sum_{j \neq k} x_{ij}\beta_j - x_{ik}\beta_k \right)^2 + \lambda \sum_{j \neq k} |\beta_j| + \lambda |\beta_k|$$

minimized at

$$\hat{\beta}_k = H_\lambda \left(\sum x_k \left(y_i - \sum_{j \neq k} x_{ij}\beta_j \right) \right)$$

One time when co-ordinate descent works!

In Code

Start at 0, update each β_k until convergence.

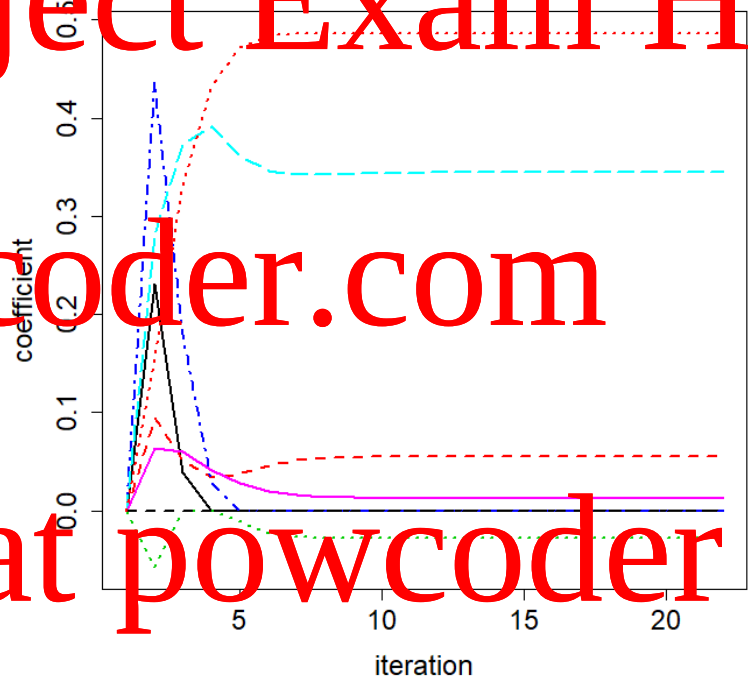
```
LASSO = function(y,X,lambda,tol=1e-8,maxit=1000){
 # Center and scale y and X
 y = scale(y); X = scale(X); n = sum(y[,1]^2)

 # Start at beta = 0
 beta = rep(0,ncol(X))
 tol.met = FALSE;  iterhist = matrix(beta,1,ncol(X)); iter=0

 while(!tol.met){
   oldbeta = beta

   # Loop over co-efficients and soft-threshold
   for(k in 1:ncol(X)){
     beta[k] = ST( t(X[,k])%*%(y - X[,-k]%*%beta[-k])/n,lambda )
   }
   iterhist = rbind(iterhist, beta); iter = iter+1

   if( max(abs(oldbeta-beta)) < tol | iter > maxit ){ tol.met=TRUE }
 }
  return(list(beta=beta, iterhist=iterhist, iter=iter) )
}
```

A Data Example

Prostate cancer volume on

- log prostate weight
- age of subject in years
- log prostatic hyperplasia
- seminal vesicle invasion
- log capsular penetration
- Gleason score
- percent Gleason 4 or 5
- prostate specific antigen

Set $\lambda = 0.05$



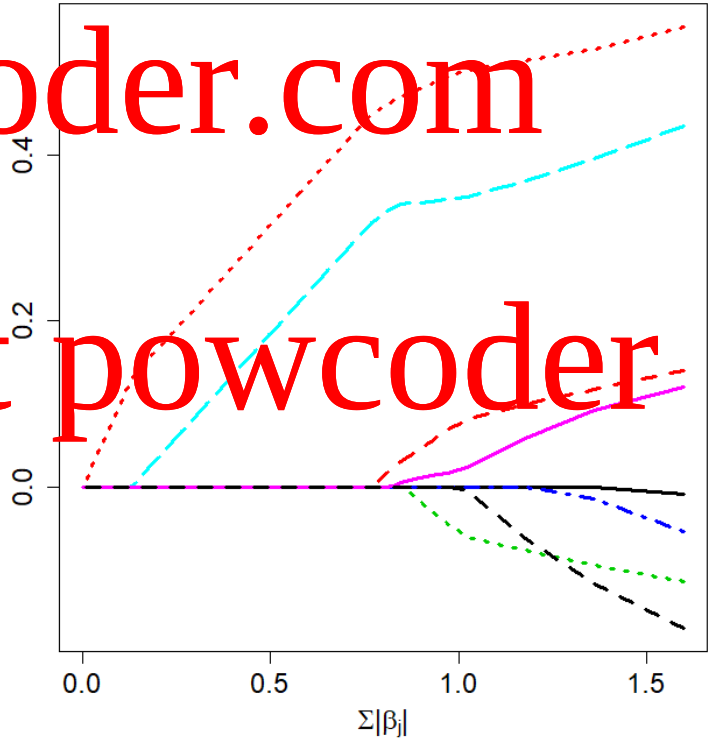

```
> lasso.result = LASSO( prostate[,1],prostate[,-1],0.05)
> lasso.result$beta
[1]  0.00000000  0.05480074 -0.02788401  0.00000000  0.34451971  0.01304833
[7]  0.00000000  0.48628871
```

Searching Over λ

```
lambdaseq = seq(0,1,by=0.01)
betamat = matrix(0,length(lambdaseq),ncol(X))

for(l in 1:length(lambdaseq)){
  betamat[l,] = LASSO(prostate[,1],prostate[,-1],lambdaseq[l])$beta
}
```

Nicest plot is β versus $\sum |\beta_j|$

```
betanorm = apply( abs(betamat),1,sum)

matplot(betanorm,betamat)
```



But still need to decide on which λ to use.

Extensions

- Non-quadratic losses:
  - Poisson regression

$$\log P(y_i|\mathbf{x}_i, \beta) = y_i\left(\sum x_{ij}\beta_j\right) - e^{\sum x_{ij}\beta_j} + \text{const}\, y_i$$

    fit with penalty

$$\sum \log P(y_i|\mathbf{x}_i, \beta) + \lambda \sum |\beta_j|$$

    - Check derivative at 0 is bigger than λ, but then need numerical optimization.
    - Also logistic regression.
- Different types of penalties or constraints
  - $\sum |\beta_j - \beta_{j+1}|$ – sequence of coefficients should be the same (fused LASSO)
  - $\sum \sqrt{\sum_{subset} \beta_j^2}$ *groups* of coefficients should all be zero (group LASSO)

  Can require more specialized methods.

Important note: no inference after LASSO; not even bootstrap.

Summary

Constraints/penalization in Statistics when

- Natural parameter ranges
- Testing particular hypotheses
- As model selection

Many procedures; not all optimization methods work well.

Penalization for model selection increasingly popular (many varieties); but we still can't do inference for it.

Next: nonparametric smoothing.