Constrained Optimization

Constrained Optimization

In many problems, there are natural constraints on optimization

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

We m

0 1

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

Also used for model selection:

$$\sum |\beta_j| \le C.$$

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

$$\frac{\theta}{\quad} \qquad \frac{\quad}{\quad} \qquad]$$

But, may change optimization curvature.

Positive Constraints

Log transformation is common

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

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

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 $\mathbf{x}$.

Sepa

-
  - 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:

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

-
- 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
-
- Can also be used for additional nonlinear constraints:

minimize $F(\mathbf{x})$

subject to $G(\mathbf{x}) \geq 0$

and $H(\mathbf{x}) = 0$

In Model Selection

In linear regression

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

whe [illegible] $\beta_j = 0$.

Rece [illegible]

minimize $\sum_{i=1}^{n} \big(y_i - \beta_0$ [illegible]

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:

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

Simp

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

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

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.

$$\frac{d}{d\beta_1}$$

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

Penalty $\lambda|\beta_1|$

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

Illustration

$\lambda = 0.2$ $\lambda = 1$

Minimum outside 0 Minimum at 0

Derivatives

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

Cha

$$_i \; _i - \frac{\qquad}{2} \qquad _i y_i)$$

otherwise the minimum is at

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

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 threshold function

(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

$$\sum_i \Big(y_i - \sum_j x_{ij}\beta_j\Big)^2 + \lambda \sum_j |\beta_j|$$

y's, x's centered, scaled.

Writ

$$\Big(y_i - \sum_{j\neq k} x_{ij}\beta_j - x_{ik}\beta_k\Big)$$

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(X[,1]^2)

 # Start at beta = 0
 beta = r
 tol.

 whil
   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
   }
   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

Set $\lambda = 0.05$

- log prostate weight
- age of subject in years
-
-

- log capsular penetration
- Gleason score
- percent Gleason 4 or 5
- prostate specific antigen


```
> 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
}
```


Nice


```
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

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

    fit with penalty

  - numerical optimization.
  - Also logistic regression.
- Different types of penalties or constraints

  - $\sum |\beta_i - \beta_{i+1}|$ – sequence of coefficie (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
-
-

Many procedures; not all optimization methods work well.

Penalization for model selection increasingly p
varieties); but we still can't do inference for t

Next: nonparametric smoothing.