Penalized Regression

Vahid Partovi Nia

Advanced Machine Learning: Lecture 01

February 10, 2018

Outline

Numerical Consideration

- Cholesky is faster than LU
- LU is faster than QR
- QR is faster than SVD

$$S(\boldsymbol{\beta}) = \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

$$\min S(\boldsymbol{\beta}) \Rightarrow (\mathbf{X}^\top\mathbf{X})\boldsymbol{\beta} = \mathbf{X}^\top\mathbf{y}$$

Suppose $\mathbf{Q}$ is an orthogonal (rotation) matrix. Then $S(\boldsymbol{\beta}) = (\mathbf{Q}\mathbf{y} - \mathbf{Q}\mathbf{X}\boldsymbol{\beta})^\top(\mathbf{Q}\mathbf{y} - \mathbf{Q}\mathbf{X}\boldsymbol{\beta})$

Algorithm

1. Decompose $\mathbf{QX} = \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix}$, where $\mathbf{R}$ is upper triangular.
2. Partition $\mathbf{Qy} = \begin{pmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \end{pmatrix} \mathbf{y}$
3. Back-solve $\mathbf{R}\boldsymbol{\beta} = \mathbf{q}_1$

Suppose $\mathbf{X}^\top\mathbf{X} = \mathbf{LU}$, $\mathbf{L}$ is lower traingular and $\mathbf{U}$ is upper triangular.

1. Decompose $\mathbf{X}^\top\mathbf{X} = \mathbf{LU}$
2. Back-solve $\mathbf{U}\mathbf{q}_1 = \mathbf{X}^\top\mathbf{y}$
3. Back-solve $\mathbf{L}\boldsymbol{\beta}_2 = \mathbf{q}_1$

Cholesky

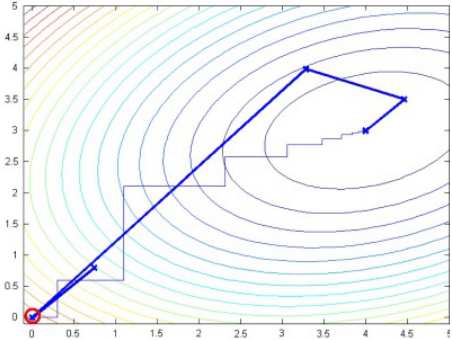
Suppose $\mathbf{X}^\top\mathbf{X} = \mathbf{A}^\top\mathbf{A}$, where $\mathbf{A}$ is lower triangular.

1. Decompose $\mathbf{X}^\top\mathbf{X} = \mathbf{A}^\top\mathbf{A}$
2. Back-solve $\mathbf{A}\mathbf{q}_1 = \mathbf{X}^\top\mathbf{y}$
3. Back-solve $\mathbf{A}^\top\boldsymbol{\beta} = \mathbf{q}_1$

Suppose $\mathbf{X} = \mathbf{PDQ}$, where $\mathbf{D}$ is diagonal-zero and $\mathbf{Q}$ is orthogonal. This means $\mathbf{X}^\top\mathbf{X}$ is symmetric.

1. Decompose $\mathbf{X}^\top\mathbf{X} = \mathbf{Q}^\top\mathbf{D}\mathbf{Q}$, where $\mathbf{D}$ is diagonal, and $\mathbf{Q}$ is orthogonal.
2. $\boldsymbol{\beta} = \mathbf{Q}^\top\mathbf{D}^{-1}\mathbf{Q}\mathbf{X}^\top\mathbf{y}$

coordinate vs conjugate



$$
\begin{aligned}
\hat{\beta}_j^{t+1} &= \operatorname{argmin} S\left\{(\hat{\boldsymbol{\beta}}_{j-1}^{t}, \boldsymbol{\beta}_j, \hat{\boldsymbol{\beta}}_{j+1}^{t})\right\} \\
\hat{\boldsymbol{\beta}}^{t+1} &= \hat{\boldsymbol{\beta}}^{t} - \delta \left\{\frac{\partial^2 S(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^\top}\right\}^{-1} \Big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}^t} \frac{\partial S(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}} \Big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}^t}
\end{aligned}
$$

coordinate descent for LS

Remember in $y_i = \beta_1 x_{i1}$ the LS estimator is Suppose

$y_i = y_i - \bar{y}$, so there is no need for $\hat{\beta}_0$.

$$\text{argmin } \frac{1}{2}\sum_i (y_i - x_{1i}\beta_1 - \ldots - x_{ij}\beta_j - \ldots - x_{ip}\beta_p)^2$$

coordinate descent for LS

Remember in $y_i = \beta_1 x_{i1}$ the LS estimator is Suppose

$y_i = y_i - \bar{y}$, so there is no need for $\hat{\beta}_0$.

$$\text{argmin}\ \frac{1}{2}\sum_i (y_i - x_{1i}\beta_1 - \ldots - x_{ij}\beta_j - \ldots - x_{ip}\beta_p)^2$$

$\hat{\beta}_j = \frac{\sum_{i=1}^n x_{ij} r_i^t}{\sum_{i=1}^n x_{ij}^2}, \quad r_i^t = ?$

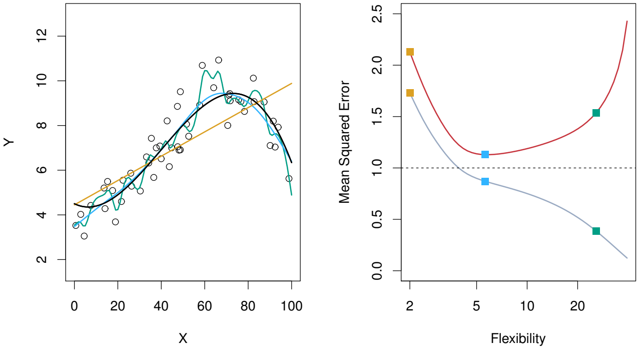
What is the difference between a predictor $\hat{\mathbf{y}}$ and an estimator $\hat{\boldsymbol{\beta}}$?

- $\mathrm{MSE}(\hat{\mathbf{y}}) = \mathbb{E}(\hat{\mathbf{y}} - \mathbf{y})^\top(\hat{\mathbf{y}} - \mathbf{y}) \approx$ cross-validation. Overfitting!
- $\mathrm{MSE}(\hat{\boldsymbol{\beta}}) = \mathbb{E}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^\top(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \approx$bootstrap

Prediction MSE

Suppose a new $\mathbf{x}_0$ is not observed in $\mathbf{x}_i, i \in \{1, \ldots, n\}$
$y_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \varepsilon_i, \mathbb{E}(\varepsilon_i) =, \mathbb{V}(\varepsilon_i) = \sigma^2$

$$\mathrm{MSE}\{\hat{y}(\mathbf{x}_0)\} = \mathbb{V}\{\hat{y}(\mathbf{x}_0)\} + \mathrm{bias}^2\{\hat{y}(\mathbf{x}_0)\} + \sigma^2$$



regplot code

Numerical Questions

Polynomial regression of y over one x

$$y_i - \bar{y} = \beta_1 x_{i1} + \ldots + \beta_k x_i^k + \varepsilon_i$$

- Why condition number of $(\mathbf{X}^\top \mathbf{X})$ is important?
- What if the smallest eigenvalue of $\mathbf{X}^\top \mathbf{X}$, say $\lambda_p \approx 0$?
- When does this happen in polynomial regression?
- How to find LS if the polynomial order $k < n$?

Numerical Questions

Polynomial regression of y over one x

$$y_i - \bar{y} = \beta_1 x_{i1} + \ldots + \beta_k x_i^k + \varepsilon_i$$

- Why condition number of $(\mathbf{X}^\top \mathbf{X})$ is important?
- What if the smallest eigenvalue of $\mathbf{X}^\top \mathbf{X}$, say $\lambda_p \approx 0$?
- When does this happen in polynomial regression?
- How to find LS if the polynomial order $k < n$?
- How to find LS if the polynomial order $k \geq n$?

Suppose the constant model $y_i = \beta_0 + \varepsilon_i$

$$
\begin{aligned}
\hat{\beta}_0 &= c\bar{y} \\
\hat{c} &= \text{argmin MSE}(c) \\
\text{MSE}(c) &= \mathbb{E}(\hat{\beta}_0 - \beta_0)^2 \Rightarrow c =?
\end{aligned}
$$

Shrunken mean

Suppose multivariate mean estimation problem of dimension $p \geq 3$ while

$$\mathbf{y}_{p\times 1} \sim \mathcal{N}(\boldsymbol{\beta}_{p\times 1}, \mathbf{I}_{p\times p}).$$

Stein showed

$$\left\{1 - \frac{(d-2)}{\sum_{j=1}^{p} y_j^2}\right\}\mathbf{y}$$

estimates $\boldsymbol{\beta}$ better than $\mathbf{y}$ in terms of MSE

Penalized mean

For a known penalization constant λ, find

$$\text{Ridge}: \ \hat{\beta}_0(\lambda) = \operatorname{argmin} \frac{1}{2}\sum_{i=1}^{n}(y_i - \beta_0)^2 + \lambda\beta_0^2$$

Penalized mean

For a known penalization constant λ, find

$$\text{Ridge}: \ \hat{\beta}_0(\lambda) = \text{argmin} \ \frac{1}{2}\sum_{i=1}^{n}(y_i - \beta_0)^2 + \lambda\beta_0^2$$

$$\hat{\beta}_0(\lambda) = \frac{n}{n+\lambda}\bar{y}$$

Penalized mean

For a known penalization constant λ, find

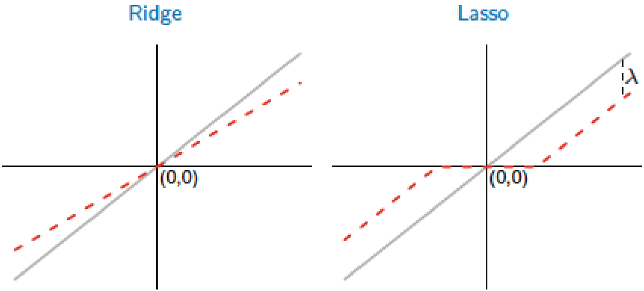
$$\text{Ridge}: \quad \hat{\beta}_0(\lambda) = \operatorname{argmin} \frac{1}{2}\sum_{i=1}^{n}(y_i - \beta_0)^2 + \lambda\beta_0^2$$

$$\hat{\beta}_0(\lambda) = \frac{n}{n+\lambda}\bar{y}$$

$$\text{Lasso}: \quad \hat{\beta}_0(\lambda) = \operatorname{argmin} \frac{1}{2}\sum_{i=1}^{n}(y_i - \beta_0)^2 + \lambda|\beta_0|$$

$$\hat{\beta}_0(\lambda) = \operatorname{sign}(\bar{y})\left\{\bar{y} - \frac{\lambda}{n}\right\}_+$$

Plot $(\bar{y}, \hat{\beta}_0(\lambda))$



Ridge Regression

$$S(\boldsymbol{\beta}) = \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^\top\boldsymbol{\beta}$$

Show that

$$\hat{\boldsymbol{\beta}} = \mathrm{argmin}\ S(\boldsymbol{\beta}) = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$$

How do you compute?

Ridge Regression

$$S(\boldsymbol{\beta}) = \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^\top\boldsymbol{\beta}$$

Show that

$$\hat{\boldsymbol{\beta}} = \text{argmin } S(\boldsymbol{\beta}) = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$$

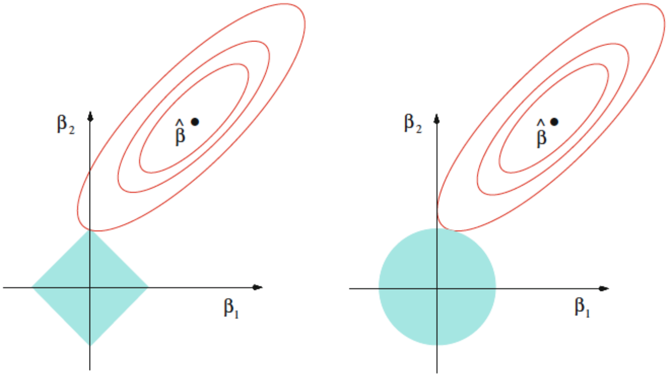
How do you compute?
using Cholesky? LU? QR? SVD? coordinate? conjugate?

Lasso

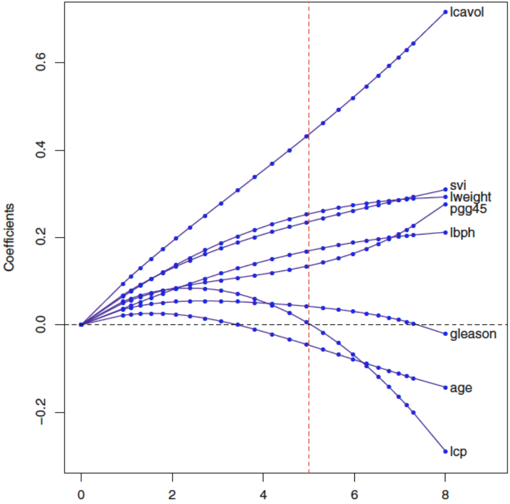
$$S(\boldsymbol{\beta}) = \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j|$$

How do you compute?

Visual penalization



Ridge coefficients



Lasso coefficients