Elements of Statistical Learing

Week 3 ~ Week 4: Linear Regression and Related Methods Part 2

앞서, penalty 항이 없는 Linear Regression에 대해서 알아보았다. 이제 penalty을 통해서 계수를 shrinkage하는 Ridge와 Lasso에 대해서 알아보자.

Ridge Regression

Ridge esimator를 formal하게 적으면 아래와 같다.

$$\begin{aligned}\hat{\boldsymbol{\beta}}^{Ridge} &= argmin_{\boldsymbol{\beta}} \left\{ (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})' (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}' \boldsymbol{\beta} \right\} \\ &= argmin_{\boldsymbol{\beta}} \left\{ \sum_i (y_i - \beta_0 - \sum_j \beta_j x_{ij})^2 + \lambda \sum_j \beta_j^2 \right\}\end{aligned}$$

기존의 OLS는 $\hat{\boldsymbol{\beta}}^{OLS} = argmin_{\boldsymbol{\beta}} \left\{ (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})' (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right\}$로 정의됐다. 즉, Ridge Estimator는 penalty를 추가한 Quantity를 minimize한다는 점이 OLS와는 다르다. 그럼 penalty의 역할이 무엇일까? 계수 β_j를 shrinkage하는 역할이다. 여기서 λ는 tuning parameter로, shrinkage의 정도를 조절한다. 예를 들어 아주 큰 λ를 생각해보자. λ가 아주 크다면 $\sum_i (y_i - \beta_0 - \sum_j \beta_j x_{ij})^2$의 중요도는 작아지고, quantity를 최소로 하기 위해, λ에 곱해진 $\sum_j \beta_j^2$를 작게 만들려고 할 것이다. 즉, penalty의 영향력이 커지고 계수들은 전체적으로 0에 shrinkage 된다. 반면에, λ가 작아서 0이라면, penalty는 없어지고 Ridge estimator는 OLS와 동일해진다. Ridge estimator는 동일하게 아래와 같이 쓸 수 있다.

$$\hat{\boldsymbol{\beta}}^{Ridge} = argmin_{\boldsymbol{\beta}} \left\{ (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})' (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right\} \ subjet \ to \ \sum_j \beta_j^2 \leq s$$

여기서 s와 λ는 one-to-one correspondence를 가진다. 즉, s가 작아진다면 제한하는 영역이 원점과 가까워지면서 줄어든다. 즉, 제한하는 정도가 강해지므로 계수들이 0에 가까워진다. 따라서 이는 λ가 커지는 것과 동일하다. 반대로, s가 커지는 것은 λ가 작아지는 것과 동일하다.

Ridge regression에서, penalty는 항상 $\sum_{j=1}^p \beta_j^2$로 쓰여진다. 뭔가 빠진 것이 있지 않은가? penalty 항에는 β_0를 포함하지 않는다. 그 이유를 살펴보자. minimize하고자 하는 quantity는 아래와 같이 다시 쓸 수 있다.

$$\begin{aligned}\left\{ \sum_i (y_i - \beta_0 - \sum_j \beta_j x_{ij})^2 + \lambda \sum_j \beta_j^2 \right\} &= \sum_i \left(y_i - \bar{y} - \sum_j \beta_j (x_{ij} - \bar{x}_j) + \left(\bar{y} - \sum_j \beta_j \bar{x}_j - \beta_0 \right) \right)^2 + \lambda \sum_j \beta_j^2 \\ &= \sum_i \left(y_i - \bar{y} - \sum_{j=1} \beta_j (x_{ij} - \bar{x}_j) \right)^2 + \lambda \sum_{j=1} \beta_j^2 + N \left(\bar{y} - \sum_{j=1} \beta_j \bar{x}_j - \beta_0 \right)^2\end{aligned}$$

이를 minimize하는 β_0의 estimator는 $\hat{\beta}_0 = \bar{y} - \sum_{j=1} \beta_j \bar{x}_j$이며 데이터를 centering한다면, 이는 항상

0이다. 조금의 유도 과정을 거치면 $\hat{\boldsymbol{\beta}}^{Ridge}$ 을 아래와 같이 도출할 수 있다. ($\mathbf{y}, \mathbf{X}$는 centering 되었음)

$$\hat{\boldsymbol{\beta}}^{Ridge} = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y}$$

Ridge estimator을 이용한 fitted value와 OLS을 이용한 fitted value을 SVD를 통해 비교하면 몇 가지 재미있는 성질을 알 수 있다. centering된 design matrix을 $\mathbf{X} = \mathbf{UDV}'$ 로 SVD한뒤 유도 과정을 거치면

$$\mathbf{X}\hat{\boldsymbol{\beta}}^{LSE} = \sum_{j=1}^{p} \mathbf{u}_j\mathbf{u}_j'\mathbf{y}$$

$$\mathbf{X}\hat{\boldsymbol{\beta}}^{Ridge} = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda}\mathbf{u}_j\mathbf{u}_j'\mathbf{y}$$

즉, $\mathbf{X}\hat{\boldsymbol{\beta}}^{LSE}$ 는 $\mathbf{U}$의 column space로 $\mathbf{y}$를 projection한 결과이다. ($\mathbf{u}_j(\mathbf{u}_j'\mathbf{u}_j)^{-1}\mathbf{u}_j' = \mathbf{u}_j\mathbf{u}_j'$) 또한 $\mathbf{X}\hat{\boldsymbol{\beta}}^{Ridge}$ 도 $\mathbf{U}$의 column space로 $\mathbf{y}$를 projection하는데, $\frac{d_j^2}{d_j^2 + \lambda}$ 라는 shrinkage effect을 가진다. 만약에 $\lambda = 0$이라면 약분이 되어, OLS가 투영하는 공간과 동일해지고, λ가 매우 커진다면 d_j^2의 값에 상관 없이 shrinkage가 많이 일어날 것이다. λ가 고정되어 있고 d_j^2가 매우 크다면 1과 가까워지므로 shrinkage effect가 크지 않고 d_j^2가 매우 작다면 $\frac{d_j^2}{d_j^2 + \lambda} \approx \frac{1}{\lambda}$가 되어 shrinkage effect가 커질 것이다.

d_j^2를 자세히 살펴보자. centering 된 $\mathbf{X}$의 sample covariance matrix을 $\mathbf{S} = \mathbf{X}'\mathbf{X}/N$이라고 두고 $\mathbf{X}'\mathbf{X}$의 eigen decomposition은 $\mathbf{X}'\mathbf{X} = \mathbf{VD^2V}'$ 이다. 즉, $\mathbf{V}$는 $\mathbf{X}'\mathbf{X}$의 eigen vector로 구성된 행렬이며, PCA에서 j번째 principal component direction은 j번째 eigen vector이다. 따라서 $\mathbf{X}\mathbf{v}_j$는 $\mathbf{X}$의 columns의 선형 결합이고, 이는 곳 j번재 principal component이다.

$$\mathbf{z}_j = \mathbf{X}\mathbf{v}_j$$

그런데, $\mathbf{X} = \mathbf{UDV}$에서 $\mathbf{XV}' = \mathbf{UD}$이고 $\mathbf{X}\mathbf{v}_j = d_j\mathbf{u}_j$이다. 따라서 $\mathbf{z}_j = \mathbf{X}\mathbf{v}_j = d_j\mathbf{u}_j$이다. 즉, $\mathbf{z}_j'\mathbf{z}_j/N = d_j^2/N$이고, d_j^2는 j번째 principal component의 sample variance랑 관련이 있다. 작은 d_j^2 값은 shrinkage effect가 크다는 뜻인데, 어디로 커지냐면 $\mathbf{X}$의 column space 내에서 작은 분산을 가지는 쪽으로 커진다는 뜻이다.

The LASSO

LASSO는 L_1 penalty을 가진다.

$$\begin{aligned}
\hat{\boldsymbol{\beta}}^{Lasso} &= argmin_{\boldsymbol{\beta}} \left\{ \sum_i (y_i - \beta_0 - \sum_j \beta_j x_{ij})^2 + \lambda \sum_j |\beta_j| \right\} \\
&== argmin_{\boldsymbol{\beta}} \left\{ \sum_i (y_i - \beta_0 - \sum_j \beta_j x_{ij})^2 \right\} \; subject\ to \; \sum_j |\beta_j| \leq s
\end{aligned}$$

Ridge에서와 마찬가지로 λ는 shrinkage를 control하는 tuning parameter이다. Ridge와는 달리, Lasso

는 $\hat{\boldsymbol{\beta}}^{Lasso}$의 closed form이 존재하지 않는다. 그리고 Ridge는 계수를 정확히 0으로 shrinkage하지 않지만, Lasso는 계수를 정확히 0으로 shrinkage한다. 따라서 Lasso는 variable selection의 역할도 수행한다.

Elastic Net

Lasso가 variable selection 기능을 한다고 하지만, 성능이 Ridge보다 좋지 않을 때가 많았고 서로 correlate된 변수들 중에서 단 하나의 변수만 선택하는 문제점이 있었다. 이러한 motivation으로 인해, Ridge와 Lasso를 혼합한 Elastic Net이 등장하였다.

$$
\begin{aligned}
\hat{\boldsymbol{\beta}}^{Enet} &= argmin_{\boldsymbol{\beta}} \left\{ \sum_i (y_i - \beta_0 - \sum_j \beta_j x_{ij})^2 + (1-\alpha) \sum_j |\beta_j| + \alpha \sum_j \beta_j^2 \right\} \\
&== argmin_{\boldsymbol{\beta}} \left\{ \sum_i (y_i - \beta_0 - \sum_j \beta_j x_{ij})^2 \right\} \ subject\ to\ (1-\alpha) \sum_j |\beta_j| + \alpha \sum_j \beta_j^2 \le s
\end{aligned}
$$