Model Selection

Why?

- Prediction Accuracy
  - Predictor의 개수가 observation의 개수보다 많아지는 경우, 하나의 observation에 대해서 세세한 학습이 이루어지는 것을 의미한다. 즉 너무 적은 양의 데이터에 대해서 학습해야 하는 정보가 너무 많고, 그에 따라서 새로운 데이터가 들어왔을 때 그 variance가 매우 높아질 수 있다는 것을 의미한다.
  - 이를 차원의 저주(Curse of dimensionality)라고 부르기도 한다.
- Model Interpretability
  - 마찬가지로, 적은 양의 데이터에 대해서 너무 많은 정보가 주어지는 경우, 불필요한 정보의 양도 그만큼 늘어난다. 따라서 유의미한 변수의 개수가 줄어들 수 있으며, 변수끼리의 관계에 따라서 정확한 추정이 이뤄지지 않는 경우도 있게 된다.
- 따라서 변수의 크기를 적당하게 유지하는 것이 중요하다.

Model Selection의 Methods

- Subset selection: 주어진 데이터의 feature들 가운데 일부분만을 선택하는 방법
- Shrinkage: Target 예측과 관련없는 변수들의 coefficient를 0에 가깝거나 0으로 만듦으로써 관련 없는 변수를 솎아내는 것
- Dimension Reduction: p-dimensional space에 대해서 그보다 작은 M-dimensional subspace로 projection하는 방법

  220414

Performance Metric for Subset selection

- 특히 for linear regression, R^2와 RSS는 subset selection에서 사용하지 않음
  - $R^2 = 1 - \frac{RSS}{TSS}$인데 $RSS = \sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2 = n \cdot MSE = n\left(\text{bias}^2 + \text{variance}\right)$
    - 그리고 변수의 개수가 늘어날 때마다 bias가 감소하므로, MSE 나아가 RSS까지 감소

        - In other words, 변수가 적당한 크기일 경우 변수의 개수 증가 → 정보량의 증가로 보다 정확한 estimation이 가능해지고, 그에 따라서 RSS는 감소하고 R^2는 증가함
    - 즉, R^2와 RSS는 모두 model size를 고려하지 않는 metric이므로 variable의 개수가 늘어남에 따라서 모두 그 성능이 개선되는 (것처럼 보이는) metric임
- 따라서 alternative metrics가 사용되고, 그 목록은 다음과 같다.
    - Adjusted-R^2
        - R^2를 전체 observation의 개수와 사용한 변수의 개수를 모두 고려하여 normalize한 metric

$$\text{Adjusted-}R^2 = 1 - \frac{RSS/\left(n-d-1\right)}{TSS/\left(n-1\right)} = 1 - \frac{RSS \cdot \left(n-1\right)}{TSS \cdot \left(n-d-1\right)}$$

        - d **increases → denominator decreases → ratio increases → Adjusted** R^2 **decreases** for the given TSS, RSS, and n
        - 나머지와 달리 유일한 higher-the-better metric
    - AIC (Akaike Information Criterion)

$$AIC = -2\log L + 2d$$

        - $\log L$은 log likelihood, d는 the number of variables
            - 특히 $-2\log L$**은 deviance**라고 불리며 RSS의 보다 광범위한 개념으로 사용된다.
        - given d에 대해서 log likelihood는 높아야 하므로 **AIC는 lower-the-better metric**
        - 그런데 d가 커질수록 설명력이 높아져 log likelihood는 높아지지만 AIC가 커질 위험이 있기 때문에 둘의 balance를 맞추는 지점에서 최적의 d를 설정
    - BIC (Bayesian Information Criterion)

$$\begin{aligned} BIC &= -2\log L + d\log n \\ &= \frac{1}{n}\left(RSS + d\hat{\sigma}^2 \log n\right) \end{aligned}$$

        - AIC와 마찬가지로 lower-the-better metric이며, $\log 7 \approx 1.95$이므로 **전체 observation이 8개 이상이기만 하다면 AIC보다 feature의 개수에 더 강한**

penalty를 부과하는 metric

- AIC처럼 설명력(represented by log likelihood)과 model size를 모두 고려해 balance가 맞는 지점에서 최적의 d를 설정

- Mallow's C_p

$$C_p = \frac{1}{n}\left(RSS + 2d\hat{\sigma}^2\right)$$

  - $\frac{RSS}{n}$은 negative log likelihood와, $\frac{2d\hat{\sigma}^2}{n}$는 $2d$와 연관
  - BIC에서도 알 수 있듯이 linear model에서는 사실상 AIC와 equivalent metric이며, 따라서 lower-the-better metric

Subset Selection

Best Subset Selection

- Theoretical한 방법으로, 모든 변수 조합에 대해서 performance metric이 가장 좋은 변수 조합을 sequentially 찾는 방법
- Steps
  - 아무 변수도 포함하지 않은 null model을 M_0라 하면, M_0은 전체 y의 평균 $\bar{y}$로서 예측한다.
  - 전체 p개의 변수에 대해
    - $k = 1, 2, \dots, p$을 돌면서 각각의 k에 대해 $\binom{p}{k}$개의 모든 변수 조합을 찾고, 각각의 변수 조합을 model에 추가하여 R^2를 계산한다.
      - 즉, $k = 1$인 경우 $\binom{p}{1} = p$개의 simple regression model이 나오게 되고, 그 가운데 가장 R^2가 높은 model을 M_1이라 한다.
      - 마찬가지로 $k = 2$에 대해서도 $\binom{p}{2} = \frac{p(p-1)}{2}$개의 multiple regression model이 나오게 되고, 그 가운데 R^2가 가장 높은 model을 M_2라 한다.
  - 이렇게 설정한 $M_1, M_2, \dots, M_p$에 대해서, model size를 고려한 performance metric이 가장 좋은 model을 최종 model로 선정한다.
- 단점
  - combination의 특성상 p가 늘어날수록 exponential하게 증가하기 때문에 computational burden이 매우 크다.

- 또한 R^2를 계산할 때 Test data가 아닌 Training data를 사용하기 때문에 predictive power 기준이 아닌 training fit 기준으로 최적의 조합이 도출되어 overfitting이 발생할 수도 있다.

Stepwise Selection

- **추가된 변수는 유지하는 상황 하에서 가장 나은 변수를 다시 추가**하는 방법
  - 이러한 방법을 greedy algorithm이라 부름
- Null model에서 시작해 변수를 하나씩 추가하는 forward selection, Full model에서 시작해 변수를 하나씩 제거하는 backward selection이 있다.

Forward Selection

- Algorithm
  - Null model에서 변수를 1개 추가하여 R^2가 가장 높은 model을 M_1이라 한다.
    - 이때 candidates of M_1은 총 p개가 나온다.
  - 나머지 $p-1$개의 변수 가운데 M_1에 다시 1개 추가한 뒤, R^2가 가장 높은 model을 M_2라 한다.
  - 다시 나머지 $p-2$개의 변수 가운데 M_2에 다시 1개 추가하여 R^2가 가장 높은 model을 M_3라 한다.
  - 이 과정을 전체 p에 대해 반복한 뒤 나온 $M_0, M_1, \dots, M_p$에 대해 performance metric을 적용해 가장 best score를 보이는 model을 최종 model로 선정한다.
- Pros and Cons
  - Pros: 총 계산량이 줄어든다.
    - Best subset selection에서 고려해야 하는 model은 총 $\sum_{k=1}^{p} \binom{p}{k} = 2^p$인 반면,
    - Forward selection에서 고려해야 하는 model은 총 $\sum_{k=0}^{p} p-k = \frac{p(p+1)}{2}$개로 dramatically 줄어들었음을 알 수 있다.
  - Cons: Best model을 찾을 수 있다는 보장이 없다.
    - 최적의 조합을 찾는 것이 아니라 “Given model”에서 best improvement를 찾는 algorithm이기 때문

Backward Selection

- Algorithm

    - Full model을 M_p라 하자.
    - p개 가운데 변수를 1개씩 제거해가면서 R^2의 감소분이 가장 큰 변수를 찾고, 해당 변수를 제거한 model을 M_{p-1}이라 한다.
    - 같은 방법으로 $p-1$개의 변수 가운데 1개씩 제거하면서 R^2의 감소분이 가장 큰 변수를 찾아 해당 변수를 제거한 model을 M_{p-2}라고 한다.
    - 이 과정을 null model에 도달할 때까지 전체 p에 대해 반복한 뒤 나온 $M_0, \dots, M_p$에 대해서 performance metric을 적용해 가장 best score를 보이는 model을 최종 model로 선정한다.
- Pros and Cons
    - Pros: Forward selection과 마찬가지로 계산량이 $\frac{p(p+1)}{2}$개로 줄어든다.
    - Cons
        - 마찬가지로 best model을 찾을 수 있다는 보장이 없으며,
        - Full model부터 시작하기 때문에 해당 full model이 fit될 수 있도록 $n > p$가 성립해야 한다.

Choosing the optimal model

- 위의 performance metric을 사용하는 approach를 **adjustment to the training error**라고 하며
    - AIC, BIC, C_p 모두 편리하지만, error variance를 다시 estimate해야 한다는 burden이 있음.
- Cross-validation 등 validation set을 이용해 **test error를 directly estimate할 수도 있다**.
    - Forward/Backward selection을 통해 도출된 $M_0, \dots, M_p$에 대해서 K-fold CV, holdout validation 등을 통해 test error를 MSE의 형식으로 바로 도출하며
    - 이렇게 할 경우 error variance가 더 이상 필요하지 않다는 장점이 있다.
- 두 방법 모두 Test error를 estimate하는 방식으로 각각을 통해 계산된 값들은 모두 “estimates of test MSE”라고 할 수 있다.
- 또는 one-standard-error rule를 사용할 수도 있다.
    - Estimated error curve를 그렸을 때 가장 낮은 error를 갖는 model에 대해서, 해당 model이 기록한 error로부터 1표준편차 이내에 속한 model 가운데 가장 size가 작은 model을 고르는 알고리즘

- How come?
  - “성능 차이가 크지 않다면 가장 간단한 모델이 좋다”

Shrinkage

- 특히 Linear model에 대해서 coefficient에 제약을 가함으로써, 또는 몇 개 coefficient를 0에 가깝게 만들어버림으로써 보다 관련 있는 변수에 가중치를 집중시키고 그를 통해 성능을 높일 수 있는 방법
- 대표적인 shrinkage 방법으로는 Ridge regression과 LASSO가 있음

Ridge Regression

- Linear model의 일종으로서, RSS에 L_2 norm을 더해준 loss를 최소화하는 $\hat{\beta}$를 찾고자 하는 model

$$\min_{\beta_1,\dots,\beta_p} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$

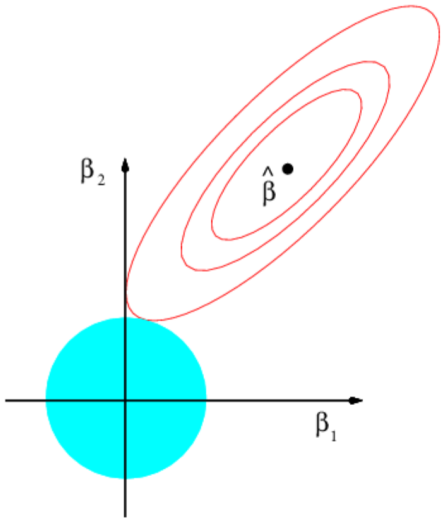
  - $\sum_{j=1}^{p} \beta_j^2 \leq C$, 즉 계수의 squared sum을 일정한 크기의 상수 C보다 작게 유지하라는 constraint 하에서 RSS를 최소화하는 $\beta_1, \dots, \beta_j$를 찾으라는 optimization problem
    - 단 constraint에 β_0은 포함되지 않으며, constraint를 λ를 붙여줌으로써 unconstrained problem으로 바꿀 수 있음
    - 원칙적으로는 $\lambda \left(C - \sum_{j=1}^{p} \beta_j^2 \right)$가 constraint로 포함되어야 하지만, β에 대해 최소화하는 데 있어 상수 C는 영향을 미치지 않으므로 생략
  - 이를 vector form으로 나타내면 아래와 같다.

$$\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - x_i \cdot \beta)^2 + \lambda ||\beta||_2^2 \right\}$$

- $\lambda \geq 0$을 tuning parameter라고 하며, λ의 크기에 따라서 constraint의 강도가 달라진다(or constraint가 차지하는 영역의 넓이가 달라진다).
  - $\lambda = 0$이면 위의 식은 RSS minimization이며, solution 역시 least square에서의 $\hat{\beta}$와 같아진다.

- $\lambda \to \infty$이면 모든 계수가 0으로 push된다.
- In general,
  - λ가 작으면 작을수록 weak constraint로, RSS는 줄어들지만 그만큼 variance가 높아진다 → overfitting 우려
  - λ가 크면 클수록 strong constraint로, variance는 줄어들지만 그만큼 RSS가 높아진다 → underfitting 우려
- 따라서 적절한 크기의 λ를 선택하는 것이 중요하며, 보통 cross-validation을 통해 validation error가 가장 작게 나오는 λ를 선택한다.

Geometric Interpretation



Ridge regression

- Red contour는 RSS에 관한 것으로, 하나의 contour line은 **같은 RSS를 갖는 모든 (β_1, β_2)의 조합**
  - $\hat{\beta} = \left(\hat{\beta}_{LS1}, \hat{\beta}_{LS2}\right)$는 Least square estimate으로서, RSS를 최소화하는 $\arg\min$이다.
  - 따라서 $\hat{\beta}$ 방향으로 갈수록 RSS는 낮아지며, $\hat{\beta}$로부터 멀어질수록 RSS가 높아진다.
  - 한편, **given dataset만을 가지고도 least square를 수행할 수 있으므로 RSS는 더 이상 움직일 수 없는 fixed curve이며 $\hat{\beta}$는 이미 계산된 constant이다.**
- Blue circle은 constraint가 차지하는 영역으로,

- Ridge regression은 L_2 norm을 사용하므로 $\beta_1^2 + \beta_2^2 \leq \frac{C}{\lambda}$라 할 수 있고, 따라서 constraint가 원의 영역을 갖게 된다.
- 그리고 λ의 크기에 따라서 constraint가 차지하는 영역의 넓이가 달라진다.
  - 커질수록 constraint가 차지하는 영역이 줄어들고(strong constraint), 작을수록 차지하는 영역이 커진다(weak constraint).

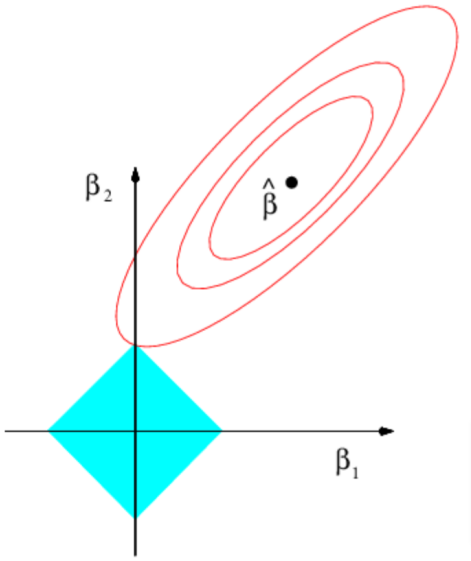
- Constrained optimization
  - Contour와 constraint가 서로 접하는 지점에서 optimal point $\left(\hat{\beta}_{R1}, \hat{\beta}_{R2}\right)$를 얻을 수 있으며,
  - $\lambda > 0$이라면 least square를 minimize하는 $\hat{\beta}$에서의 RSS보다 높은 RSS가 나올 수밖에 없다.
  - 즉, **"RSS(bias)를 어느 정도 포기하더라도 보다 robust(low-variance)한 model을 찾겠다"**는 의미

LASSO

- Ridge와 달리 L_1 norm을 constraint로 부과한 linear model

$$\min_{\beta_1,\ldots,\beta_p} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$
$$= \min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - x_i \cdot \beta \right)^2 + \lambda ||\beta||_1 \right\}$$

Geometric Interpretation



LASSO

- 마찬가지로 Red contour는 RSS curve로서, 각각의 line은 같은 RSS를 갖는 (β_1, β_2)의 조합이며,
  - $\hat{\beta}_{LS}$는 RSS를 최소화하는 least square solution
- Blue square는 constraint가 차지하는 영역으로, L_1 norm에 따라서 $|\beta_1| + |\beta_2| \leq \frac{C}{\lambda}$로 정의된다.
  - Ridge와 마찬가지로 λ가 커질수록 constraint 영역이 줄어들고(strong constraint), λ가 작아질수록 constraint 영역이 넓어진다(weak constraint).
- 역시 contour와 constraint가 만나는 점에서 optimal point를 얻을 수 있다.

Ridge vs LASSO

- Ridge는 $\lambda \to \infty$인 경우를 제외하면 coefficient를 완전히 0으로 줄일 수 없다.

$$L_{Ridge} = (y - X\mathrm{B})^T (y - X\mathrm{B}) + \lambda ||\mathrm{B}||_2^2$$

$$\frac{\partial L_{Ridge}}{\partial \mathrm{B}} = -2X^\top y + 2X^\top X\mathrm{B} + 2\lambda\mathrm{B} = 0$$
$$\Leftrightarrow \left(X^\top X + \lambda I\right)\mathrm{B} = X^\top y$$
$$\therefore \mathrm{B} = \left(X^\top X + \lambda I\right)^{-1} X^\top y$$

  - 따라서 개별 coefficient는 $\beta_j = \frac{x_j^T y}{x_j^T x_j + \lambda}$에 의해서 계산되며, 이 값은 $\lambda \to \infty$가 되지 않는 한 0이 되지 않는다.

- 반면 LASSO는 적당한 크기의 λ에 대해서도 특정한 coefficient를 완전히 0으로 shrink 할 수 있다.

$$L_{LASSO} = (y - X\mathrm{B})^{\top} (y - X\mathrm{B}) + \lambda ||\mathrm{B}||_1$$

  - $\mathrm{B} > 0$인 경우에 대해서 생각해보면

$$\begin{aligned} &\frac{\partial L_{LASSO}}{\partial \mathrm{B}} = -2X^{\top}y + 2X^{\top}X\mathrm{B} + \lambda = 0 \\ &\Leftrightarrow \left(X^{\top}X\right)\mathrm{B} = X^{\top}y - \frac{\lambda}{2} \\ &\therefore \mathrm{B} = \left(X^{\top}X\right)^{-1}\left(X^{\top}y - \frac{\lambda}{2}\right) \end{aligned}$$

  - 따라서 적절한 λ를 찾을 수 있다면 개별 coefficient를 완전히 0으로 만들 수 있다.
- 이러한 특징으로 인해 LASSO yields sparse model이라고 하며, LASSO를 활용하면 variable selection이 가능하다.
  - 해당 계수가 완전히 0인 변수들을 제거하면 된다.
- 둘 가운데 어떤 것이 better model인지에 대해서는 주어진 데이터에 따라서 달라지지만, variable selection의 필요성이 있는 경우 LASSO를 적용하는 것이 낫다.

Scale

- Least square의 $\hat{\beta}_j$는 아래와 같이 계산된다.

$$\hat{\beta}_j = \frac{\sum_{i=1}^{n}\left(x_{ij} - \bar{x}_j\right)\left(y_i - \bar{y}\right)}{\sum_{i=1}^{n}\left(x_{ij} - \bar{x}_j\right)^2}$$

  - 따라서 x_j의 scale이 coefficient에 반영되어 있으며, scale 변화에 따라서 $\hat{\beta}_j$의 값이 바뀌게 된다.
- Coefficient의 전체 크기를 일정 수준으로 제약하는 LASSO와 Ridge에서 이는 큰 문제가 될 수 있다.
  - ex. 특정 변수의 scale이 작아 $\hat{\beta}$가 크게 계산된 경우, 해당 계수로 인해서 나머지 변수의 계수가 모두 0으로 만들어질 수 있으며, 그에 따라서 variable selection에 bias가 발생할 수 있다.
- 따라서 LASSO/Ridge를 수행할 경우 반드시 개별 feature에 대해서 sample standard deviation으로 나눠주는 normalization을 사전에 수행해야 한다.

- De-mean은 선택사항이며, normalization을 통해 모든 feature를 unit standard deviation으로 만들어야 한다.