Yonsei University, Department of Applied Statistics

Chapter 6 Linear Model and Regularization

YBIGTA Science Team 신보현

February 14, 2019

Contents

6. Linear Model Selection and Regularization

해당 챕터에서는 linear model의 연장 모델을 배운다. non-linear model을 나중에 배우겠지만 그 이전에 alternative fitting procedures을 배운다. alternative fitting procedures는 더 나은 정확도와 모델 해석력을 가지고 있다.

* Prediction Accuracy: 반응변수와 예측변수간에 true relationship이 근사적으로 선형이라는 가정 하에, least squares 추정치는 낮은 bias를 가질 것이다. 그리고 $n >> p$라면 least squares estimates는 낮은 분산을 가지고 그에 따라서 test observations에 잘 작동할 것이다. 분산이 작기 때문에 새로운 관측치에 대해서도 변동이 그렇게 크지 않기 때문이다. 만약 n이 p보다 엄청 크지 않다면 LSE추정치에 많은 variability가 있을 것이고 그에 따라 overfitting과 좋지 않은 predictions을 가지게 된다(높은 분산의 상황) 만약 $p > n$ 이라면 유일한 LSE를 얻을 수 없고 분산은 infinite해서 사용될 수가 없다. LSE로부터 얻은 계수를 constraining 하거나 shrinking 함으로써 무시할만한 정도의 bias 증가 대신, 분산을 매우 낮출 수 있다.

* Model Interpretability: 보통 다중선형회귀 모델에서 사용되는 여러 변수들이 반응변수와 관련되지 않는 경우가 많다. 관련이 없는 변수들을 포함하는 것은 모델에서 불필요한 복잡성을 가져온다. 이러한 계수들을 없애므로써 더 쉽게 해석되는 모델을 얻을 수 있다. 해당 챕터에서 자동적으로 feature selection 또는 variable selection을 해주는 모델에 관해서 공부한다.

해당 챕터에서 세개의 중요한 방법을 배운다

* Subset Selection: 이 방법은 p개의 반응변수와 관련있다고 생각되는 p개의 예측변수(reduced set of variables)를 사용하여 모델을 적합시킨다

* Shrinkage: 이 방법은 모든 p개의 predictors을 포함하는 모델을 적합한다. 하지만 LSE에 의하여 계수들은 거의 0에 가까워진다. 이러한 shrinkage(regularization이라고도 불림)은 분산을 줄이는 효과가 있다. Lasso에서의 shrinkage는 어떠힌 계수를 0으로 만든다. 따라서 shrinkage 방법도 variable selection으로 사용 될 수 있다.

* Dimension Reduction: 이 방법은 p개의 예측변수을 M 차원의 부분공간(subspace)로 project 하는 것을 포함한다. 여기서 M < p 이다. 이것은 M개의 다른 linear combinations 또는 projections을 계산함으로써 성사된다. 그리고 이 M개의 projections는 linear regression을 적합하는 predictors로 사용된다(PCR 얘기, 추후에 자세히 나옴)

6.1 Subset Selection

6.1.1 Best Subset Selection

best subset selection을 하기 위해서, p개의 predictors의 가능한 조합에서 각각 LSE regression을 실시한다. 다시 말해서, p개의 변수가 있으면, 하나의 predictor만 포함하는 p개의 모델을 적합하고, 2개의 predictors을 포함하는 ${}_pC_2$개의 모델을 적합시키고 계속 이런 식으로 한다. 그리고 모든 모델을 보는데, 이 중에서 가장 좋은것을 고른다.

Algorithm 6.1 *Best subset selection*

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

알고리즘 6.1을 보자. 변수가 p개가 있다면 변수 1개를 포함하는 모델은 p개가 있다. 우선, 변수 1개를 포함하는 p개의 모델 중 가장 좋은 모델을 선택한다. 이 때의 기준은 가장 작은 SSE 또는 동일하게 가장 큰 결정계수로 한다. 변수 2개를 포함하는 ${}_pC_2$개의 모델에 대해서도 동일한 과정을 반복한다. 그렇게 변수를 0개,1개,$\cdots$,p개 포함하는 모델에서 가장 좋은 $p+1$개의 모델이 나올 것이고 여기에서 3번의 기준을 적용하여 전체적으로 가장 좋은 모델을 선택한다. 마지막에 가장 좋은 모델을 선택할 때 SSE나 결정계수를 사용하지 않는 이유는 무엇일까? 앞서 살펴보았듯이, 변수를 더 추가함에 그 변수가 의미있는 변수이든 아니든 SSE는 지속적으로 작아진다. 따라서 SSE나 결정계수로 최종 모델을 선택하면 가장 많은 변수를 포함하는 모델을 선택하게 될 것이다.
best selection이 간단하고 개념적으로 좋은 것 같지만 그것은 computational 한계가 있다. 고려해야하는 모델의 수는 p가 늘어남에 따라서 급속하게 늘어난다. p가 40 주변에만 된다면 현대의 컴퓨터로도 계산이 힘들어진다. 따라서 다음에는 이러한 컴퓨터 계산에 효과적인 대안을 제시한다.

6.1.2 Step wise Selection

best model selection에서의 수 많은 계산은 overfitting과 높은 variance로 이어질 수 있다. 무엇 보다도, 실제 만나는 데이터들은 칼럼의 개수가 매우 많은데, 이러한 경우에 계산의 시간이 정말 오래 걸릴 것이다. 이러한 이유 때문에 restricted set of models을 지향하는 stepwise methods가 좀더 매력적인 대안이다.

Forward Stepwise Selection

Forward Stepwise Selection은 컴퓨터 상으로 best subset selection보다 효과적인 대안이다. best subset selection은 모든 2^p 개의 모델을 고려하지만 forward stepwise는 훨씬 더 적은 모델을 고려한다. 이 방법은 predictors가 없는 모델에서 시작하여 한 번에 하나의 predictors을 추가하고 이 과정을 모든 변수가 모델에 있을 때까지 반복한다. 특별히, 각 단계에서 모델에 대단한 성능을 추가하는 변수를 추가한다. 좀더 formal하게는 다음과 같은 알고리즘을 가진다.

Algorithm 6.2 *Forward stepwise selection*

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   (a) Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   (b) Choose the *best* among these $p-k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

null model에서는 p개의 변수를 하나 하나 추가해 봄으로써 얻는 모델을 비교해본다. 한 개의 변수를 포함했을 때, 가장 좋은 결과를 내는 변수를 포함시켜 변수가 한 개인 모델에서 이제 p-1개의 변수를 하나 하나 추가해봄으로써 얻는 모델을 비교해본다. 이렇게 k번째 iteration에서는 p-k개의 모델을 적합시키는 것이고 이러한 p번의 iteration을 함으로써 $\sum_{k=0}^{p-1}(p-k) = 1 + p(p+1)/2$개의 모델을 적합시킨다. 이것은 best subset selection과 비교했을 때와는 현격히 낮은 계산량이다.
하지만 이러한 계산상의 문제가 없다 하더라도 forward stepwise seletion이 항상 가장 좋은 모델을 가져오리라는 보장은 없다. 예를 들어서 변수가 3개일 때, best subset selection에서는 one-variable model은 X_1을 포함하는 것이라면 two-variable model에서는 대신 X_2, X_3을 포함한다고 하자. 그런데 forward stepwise selection은 이미 M_1에서 X_1을 이미 포함하기 때문에 M_2에서는 X_1을 무조건 포함해야 한다. 따라서, 사실은 X_2, X_3의 변수를 포함하는 것이 가장 좋지만, 변수를 쌓아가는 forward stepwise selection의 특징 때문에 이를 놓칠 수 있는 단점이 있다.

Backward Stepwise Selection

forward와는 달리 backward은 모든 변수를 포함한채로 시작하고 가장 덜 유용한 predictor를 삭제하는 방식으로 진행한다. 구체적인 알고리즘은 다음과 같다.

Algorithm 6.3 *Backward stepwise selection*

1. Let $\mathcal{M}_p$ denote the *full* model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   (b) Choose the *best* among these k models, and call it $\mathcal{M}_{k-1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

forward와 마찬가지로 backward은 $1+p(p+1)/2$개의 모델을 적합시켜 best subset selection과 비교했을 때 계산상의 장점이 있지만 가장 좋은 모델을 찾지 못할 수도 있다.

Hybrid Approaches

이 방법은 forward와 backward의 방법은 섞은 것으로써, 변수를 추가하다가 그것의 효과가 미미하다고 판단되면 변수를 삭제하는 방법이다. 이것은 best subset selection을 따라함과 동시에 forward와 backward의 계산적인 이점을 포함하려는 시도이다.

6.1.3 Choosing the Optimal Model

best, forward, back은 모두 각각 p개의 예측변수를 포함하는 모델을 만들어낸다. 여기서 가장 좋은 모델이 무엇인지 결정해야 한다. 앞에서도 말했 듯이, 모든 변수를 포함하는 모델은 가장 낮은 SSE와 가장 높은 R^2을 가지는데, 이는 이 둘이 훈련 오차와 관련되어 있기 때문이다. 보통, 미지의 데이터에 대해서 작동이 잘 되는 모델이 더 좋기 때문에, 훈련오차보다는 낮은 테스트 오차을 가지는 모델을 원한다. 따라서 이 둘은 적절한 지표가 되지 못한다. 테스트 오차와 관련해서 가장 좋은 모델을 고르려면 두 가지 방법이 있다.

1. We can indirectly estimate test error by making an adjustment to the training error to account for the bias due to overfitting

2. We can directly estimate the test error, using either a validation set approach or a cross-validation approach, as discussed in Chapter 5.

이 두 가지 접근법을 모두 살펴보도록 하자.

C_p, AIC, BIC and Adjusted R^2

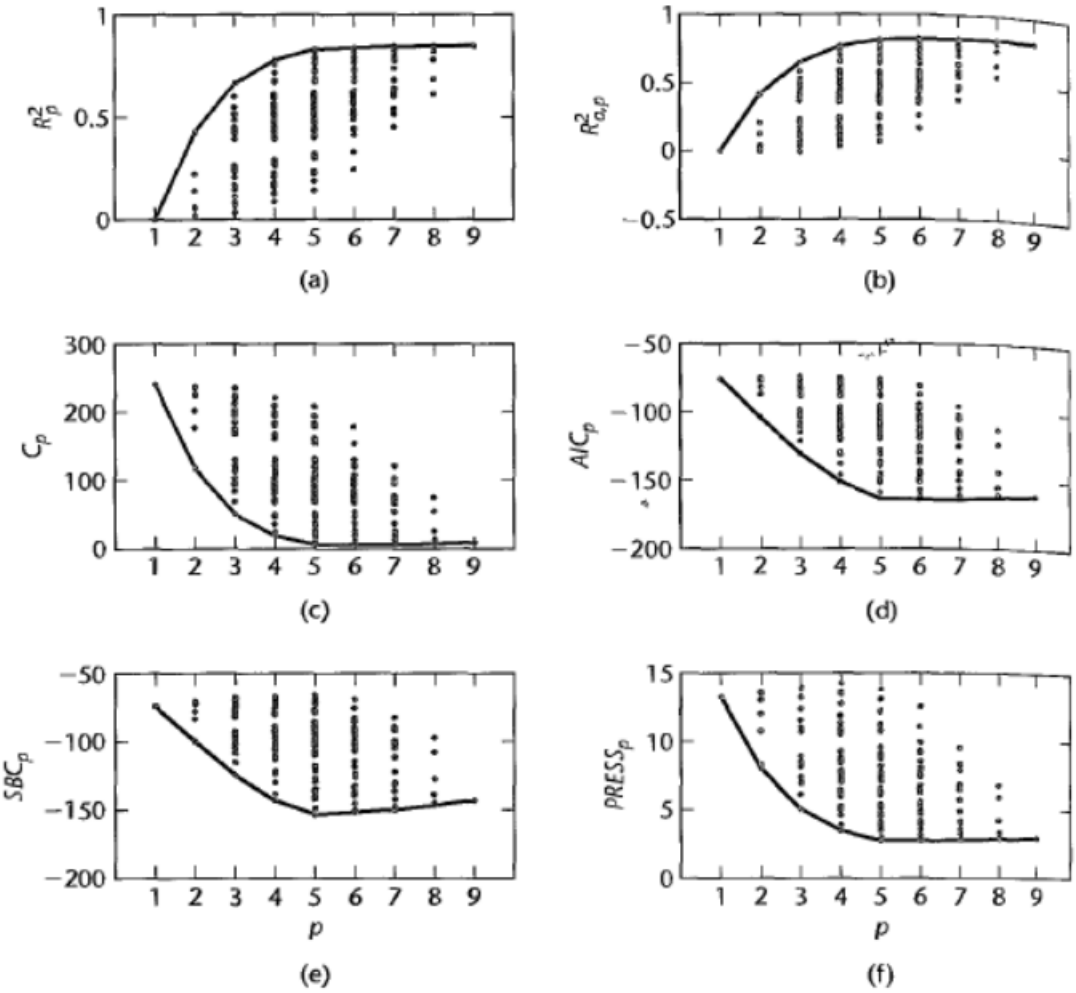
chapter 2에서 훈련세트 MSE가 일반적으로 테스트 세트 MSE를 과소평가(underestimate)한다라는 것을 배웠다. 이는 모델을 훈련데이터에 LSE를 통해 적합시킬 때, 훈련세트 MSE가 가장 작게 회귀 계수를 추정하기 때문이다. 특히 변수가 늘어남에 따라서 훈련 오차는 낮아지지만 테스트 오차는 그렇지 않을 것이다. 따라서 훈련 세트 MSE와 R^2는 다른 수의 변수를 가지고 있는 모델을 고르는데 사용될 수 없다.

회귀분석에서 변수가 증가함에 따라서 SSE가 감소함을 수식적으로 보이자.

$$\begin{aligned}
SSE(1, X_1, \cdots X_{p-1}) &= \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 X_1 - \cdots - \hat{\beta}_{p-1}X_{p-1}) \\
&= \min_{\beta_1, \cdots \beta_{p-1}} \sum_{i=1}^{n}(y_i - \beta_o - \beta_1 X_1 - \cdots \beta_{p-1}X_{p-1}) \\
&= \min_{\beta_1, \cdots \beta_{p-1}, \beta_p = 0} \sum_{i=1}^{n}(y_i - \beta_o - \beta_1 X_1 - \cdots - \beta_{p-1}X_{p-1} - \beta_p X_p) \\
&\geq \min_{\beta_1, \cdots \beta_{p-1}, \beta_p} \sum_{i=1}^{n}(y_i - \beta_o - \beta_1 X_1 - \cdots - \beta_{p-1}X_{p-1} - \beta_p X_p) \\
&= SSE(1, X_1, \cdots, X_p)
\end{aligned}$$

제약하에서 최소값을 구하는 것은 한정된 공간에서 최소값을 구하는 것이기 때문에, 실제 최소값보다 크거나 같을 수밖에 없다. 따라서 변수가 추가 됨에 따라서 그 변수가 의미가 있든 없든 SSE는 감소할 수밖에 없는 것이다.
그렇다면 변수의 개수가 다른 모델에 관해서 이들을 비교하는 지표는 무엇일까?

- Adjusted R^2(수정된 결정계수): $R^2_{a,p} = 1 - \dfrac{SSE_p(n-p)}{SSTO/(n-1)} = 1 - \dfrac{MSE_p}{SSTO/(n-1)}$
  수정된 결정 계수가 최대가 되거나, 안정화되기 시작하는 변수들을 선택하자.
- Mallow' C_p: $C_p = \dfrac{SSE_p}{MSE(X_1, \cdots, X_{r-1})} - (n - 2p)$
  작고 p(변수의 개수)에 가까운 C_p가 좋다.
- AIC_p, BIC_p: $AIC_p = nlogSSE_p - nlogn + 2p$, $BIC_p = nlogSSE_p - nlogn + plogn$
  작은 값이 좋다.
- PRESS: $\sum_{i=1}^{n}(Y_i - \hat{Y}_{i(i)})^2 = \sum_{i=1}^{n}\left(\dfrac{Y_i - \hat{Y}_i}{1 - h_{ii}}\right)^2$ where $\hat{Y}_{i(i)}$ denotes the ith fitted value without ith value is omitted, h_{ii} is diagonal element in Hat matrix
  작은 PRESS 값이 좋다.



위와 같이 변수의 개수에 따라서 각 지표의 값을 비교한 후, 적절한 변수의 조합을 구하면 된다. 예를 들어서, (b)의 그림이 수정 결정계수인데, 세로로 찍혀있는 점이 변수의 개수가 축에 써져있는 숫자와 같을 때의 여러 모델들에 대한 수정 결정계수이다. 세로에서 가장 높은 점이 동일한 변수 개수일 때, 가장 높은 수정 결정계수를 보이는 모델이다. 이를 이어보면 (b)의 곡선과 같다. 여기서 최대가 되거나, 안정화가 되기 시작하는 지점은 5개정도이다. 그 이후로는 크게 증가하지도 않으므로 변수를 추가하면 모델의 복잡성만 증가되니 변수를 5개로 선택하자.

Validation and Cross-Validation

위에 언급된 접근들에 대한 대안으로써, 각 모델들의 테스트 오차를 k-fold cross validation을 통해 추정할 수 있다. 그리고 이 추정치가 가장 낮은 모델을 최종 모델로 선택하게 된다.

이러한 방법은 테스트 오차를 직접적으로 추정한다는 것과 true underlying model에 대해서 적은 가정을 한다는 장점을 지닌다.

6.2 Shrinkage Methods

subset selection methods는 subset of predictors(예측변수의 일부)을 포함하는 선형모델을 적합하기 위해 최소자승추정법을 사용한다. 이것에 대한 대안으로, 모든 p개의 예측변수를 포함하여 모델을 적합하는데, 대신 계수를 constrains하거나 regularizes하는 기법이 있다. 왜 굳이 모든 변수를 포함하며 이러한 그에 대한 계수에 제약을 둘까? 그 이유는 shrinking methods가 분산을 현저하게 줄여주기 때문이다. 여기서는 해당 기법으로 ridge regression과 lasso을 소개한다.

6.2.1 Ridge Regression

선형회귀에서는 $\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2$을 최소화하는 $\beta_0,\cdots,\beta_p$을 적합시킨다. Ridge regression은 계수들이 살짝 다르게 최소화되는 것을 제외하고는 최소자승추정법과 거의 비슷하다. ridge regression에서는

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=Q(\beta)+\lambda\sum_{j=1}^{p}\beta_j^2 \tag{6.5}$$

을 최소화하는 $\hat{\beta}^R$을 찾는다. 여기서 $\lambda\geq 0$ 은 tuning parameter이다. $\lambda\sum_{j=1}^{p}\beta_j^2$은 shrinkage penalty로써, $\beta_1,\cdots,\beta_p$가 0에 가까울 때, 0과 가까워진다. 따라서 그것은 β_j의 estimates을 0으로 shrinking하는 효과를 가진다.

tuning parameter λ는 $Q(\beta)$와 shrinkage penalty가 regression coefficient estimates에 끼치는 영향을 조절한다. 만약 $\lambda=0$ 이라면 penalty은 아무 영향이 없고 ridge regression은 최소자승추정치와 동일한 결과를 낼 것이다. 하지만 $\lambda\longrightarrow\infty$라면 shrinkage penalty의 영향은 늘어날 것이고 전체적으로 최소화해야 하는 quantity가 무한대로 커짐에 따라서 이를 최소화하기 위해 ridge regression 계수 추정치는 0에 가까워질 것이다. 즉, 이때는 데이터가 뭐든 간에 계수 추정치를 0으로 만든다. 하나의 결과만 내는 least squares와는 달리, ridge는 λ의 값에 따라서 여러개의 계수 추정치를 생산할 것이다. 좋은 λ를 선택하는 것은 중요하다. 이것은 6.2.3에서 cross-validation으로써 얘기한다.

6.5는 equivariant하지 않다. 따라서 ridge regression의 해를 구하기 이전에 input data인 X를 centering 해준다. 위 식에서 절편에 대한 shrinkage penalty가 빠져있음을 주목하자. 절편에 대해서 shrinkage penalty을 준다면, 마치 각 반응변수 y_i에 상수 c를 더하는 것과 마찬가지인데, ridge regression의 추정치는 equivariant하지 않기 때문에 예측변수가 동일하게 c만큼 증가하지 않는다. 따라서 절편에 대한 shrinkage penalty가 없는 것이다.

절편에 대한 penalty의 내용은 ESL의 exercise 3.5에 있으며 풀이는https://stats.stackexchange.com/questions/322101/l2-normalization-of-ridge-regression-punish-intercept-if-not-how-to-solve를 참고하면 된다.

Chapter 3에서 다뤘던 최소자승 추정치는 scale equivariant하다. 즉, X_j에 상수 c를 곱함으로써 최소자승 추정치에 1/c의 scaling을 하는 결과와 같다. 다시 말해서, j번째 예측변수가 얼마나 scale되든지, $X_j\hat{\beta}_j$는 동일하게 유지 될 것이다. 이와 반면에, ridge 계수 추정치는 예측변수에 상수를 곱한다면 상당히 변화할 것이다. 따라서 ridge regression을 할 때에는 예측변수를 표준화한 후에 진행하는 것이 좋다.

$$\tilde{x_{ij}} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x_j})^2}} \tag{6.6}$$

위의 내용에서 최소자승 추정치가 scale equivariant하다는 뜻이 궁금해서 아래와 같이 정리해보았다. 다중선형회귀의 상황을 생각해보자.

$$assume\, model\, \mathbf{y} = \mathbf{X}\beta + \epsilon\, where\, \mathbf{y} \in \mathbb{R}^n, \mathbf{X}\, is\, n \times (p+1)\, matrix\, \beta \in \mathbb{R}^{p+1} and\, \epsilon\, satisfies\, assumptions$$

여기서 최소자승 추정치는 $\widehat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$이다. design matrix인 $\mathbf{X}$는 $[x_1\ x_2 \cdots x_{p+1}]$로 구성되어 있다. 여기서 x_k 칼럼에 0이 아닌 상수를 곱해서 다시 구성한 design matrix을 $\tilde{\mathbf{X}}$라 하면,

$$\mathbf{X}\begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & \cdots & & & & \\ & & & k & & & \\ & & & & \cdots & & \\ & & & & & 1 & \\ & & & & & & 1 \end{bmatrix} = [x_1 \cdots cx_k \cdots x_{p+1}] = \tilde{\mathbf{X}} = \mathbf{XS}$$

변형된 design matrix을 통해서 최소자승 추정치를 다시 구해보면 $\widehat{\beta}_{\tilde{\mathbf{X}}} = (\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}\mathbf{y}$이다. 이의 구체적인 모습을 다시 살펴보자.

$$\begin{aligned} \widehat{\beta}_{\tilde{\mathbf{X}}} &= (\tilde{\mathbf{X}}'\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}'\mathbf{y} \\ &= ((\mathbf{XS})'\mathbf{XS})^{-1}\mathbf{S}'\mathbf{X}'\mathbf{y} \\ &= (\mathbf{S}'\mathbf{X}'\mathbf{XS})^{-1}\mathbf{S}'\mathbf{X}'\mathbf{y} \\ &= \mathbf{S^{-1}}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{S^{-1}}\mathbf{S}'\mathbf{X}'\mathbf{y} \ \because \mathbf{S}\, is\, diagonal\, matrix \end{aligned}$$

여기서 $(\mathbf{X}^{'}\mathbf{X})^{-\mathbf{1}} = [z_1 \cdots z_k \cdots z_{p+1}]$ 이라 두면,

$$\mathbf{S^{-1}}(\mathbf{X}^{'}\mathbf{X})^{-\mathbf{1}} = \begin{bmatrix} z_1 & & & & & & \\ & z_2 & & & & & \\ & & \cdots & & & & \\ & & & \frac{1}{c}z_k & & & \\ & & & & \cdots & & \\ & & & & & z_p & \\ & & & & & & z_{p+1} \end{bmatrix}$$

$$\mathbf{S^{-1}}(\mathbf{X}^{'}\mathbf{X})^{-\mathbf{1}}\mathbf{S^{-1}} = \begin{bmatrix} z_1 & & & & & & \\ & z_2 & & & & & \\ & & \cdots & & & & \\ & & & \frac{1}{c^2}z_k & & & \\ & & & & \cdots & & \\ & & & & & z_p & \\ & & & & & & z_{p+1} \end{bmatrix}$$

$$\therefore \widehat{\beta}_{\tilde{\mathbf{X}}} = \mathbf{S^{-1}}(\mathbf{X}^{'}\mathbf{X})^{-\mathbf{1}}\mathbf{S^{-1}}\mathbf{S}^{'}\mathbf{X}^{'}\mathbf{y}$$

$$= \begin{bmatrix} z_1 & & & & & & \\ & z_2 & & & & & \\ & & \cdots & & & & \\ & & & \frac{1}{c^2}z_k & & & \\ & & & & \cdots & & \\ & & & & & z_p & \\ & & & & & & z_{p+1} \end{bmatrix} \begin{bmatrix} x_1^{'}y \\ x_2^{'}y \\ \cdots \\ cx_k^{'}y \\ \cdots \\ x_p^{'}y \\ x_{p+1}^{'}y \end{bmatrix}$$

$$= \begin{bmatrix} z_1x_1^{'}y & & & & & & \\ & z_2x_2^{'}y & & & & & \\ & & \cdots & & & & \\ & & & \frac{1}{c}z_kx_k^{'}y & & & \\ & & & & \cdots & & \\ & & & & & z_px_p^{'}y & \\ & & & & & & z_{p+1}x_{p+1}^{'}y \end{bmatrix}$$

한 예측 변수에 c를 곱함으로써, 그때 계수의 추정치로 원래 최소자승추정치의 1/c배가 나온 것을 확인할 수 있다. ridge regression에서는 scale equivariant하지 않다고 했는데, 왜 이런 결과가 나오는 것일까? ridge regression에서의 minimum quantity을 행렬식으로 써보면 아래와 같다.

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

이때의 해를 구해보면, $\widehat{\beta}^{Ridge}=(\mathbf{X}^{'}\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^{'}\mathbf{y}$이다. 최소자승 추정치와는 달리, 역행렬 안에 $\lambda\mathbf{I}$가 더해져 있어서 scale equivariant하게 나오지 않을 듯하다.

Why Dose Ridge Regression Improve Over Least Squares?

ridge의 장점은 bias-variance trade-off 관계에 있다. λ가 증가할수록 ridge의 flexibility는 감소하고 작은 variance와 높은 bias를 가져온다. λ가 증가한다는 것은 penalty항의 영향력이 커진다는 뜻이고 이를 최소화하기 위해서 계수에 대한 추정치는 점점 작아질 것이다. 그에 따라서 분산은 작아지고(=모델의 flexibility는 감소하고) bias는 높아진다.

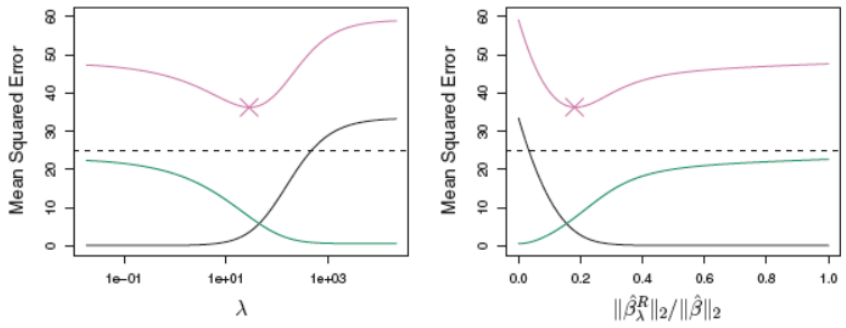


FIGURE 6.5. *Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set, as a function of* λ *and* $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$. *The horizontal dashed lines indicate the minimum possible MSE. The purple crosses indicate the ridge regression models for which the MSE is smallest.*

왼쪽그림에서 $\lambda=0$일때는 LSE로 추정한 회귀계수, 즉 일반적인 선형회귀를 실시했을 때 결과이다. 이 때는 bias는 0이고 variance는 매우 높다. λ가 커질수록(λ가 커진다는 것은 penalty가 커진다는 것이고 그렇다면 모델의 flexibility은 줄어들 것이다.) variance는 확연히 낮아지지만 bias는 그에 따라서 많이 증가하지는 않음을 볼 수 있다.
test mean squared error(test MSE)는 variance와 bias의 제곱 합이라는 것을 기억해보자. 왼쪽 그림에서는 λ가 10 이전 까지는 variance가 급격하게 낮아지는 반면 bias는 천천히

증가함으로써 Test MSE는 낮아지지만 어느 한 순간을 지나면 variance는 천천히 낮아지고 bias는 급격하게 증가하여 test MSE는 급격하게 올라간다. 이러한 일반적인 bias-variance trade-off가 아닌 ridge에서는 bias가 더 적게 증가하기 때문에 이러한 점이 ridge의 장점이라는 것이다.

사실 ridge regression 추정치는 불편추정량이 아니다. biased한 추정량이지만 대신 분산이 매우 작아지기 때문에 상황에 따라서 OLS 추정량보다 더 좋은 성능(더 작은 MSE)을 보이는 것이다.

오른쪽 그림은 x축이 ridge의 ℓ_2 norm을 least squares의 ℓ_2 norm으로 나눈 값이다. 여기서 $\|\beta\|_2 = \sqrt{\sum_{j=1}^{p} \beta_j^2}$으로 정의되고 $\|\beta\|_2$이 β의 norm이라고 보면 된다. 이제 오른쪽 그림에서 왼쪽에서 오른쪽으로 갈 수록 더 flexible하게 되고 bias는 줄어들고 variance는 증가한다. 일반적으로 반응변수와 예측변수의 관계가 선형에 가깝다면 최소자승 추정치는 bias는 낮지만 높은 variance을 가진다. 이것은 training data에서의 작은 변화가 최소자승 추정치에 큰 변화를 가져온다는 뜻이다. 또한 만약 p가 n에 거의 가깝다면 최소자승 추정치는 매우 변동적일 것(데이터가 적으면 적을수록 과적합의 위험이 있고 이는 큰 분산을 가지는 상황이다)이고 $p > n$이라면 하나의 해를 가지지 않을 것이다. 하지만 $p > n$인 상황에서도 ridge regression은 해를 가지고 분산을 매우 많이 줄이는 대신 bias가 조금 증가하게 되어서 나름 좋은 추정치를 만든다.

6.2.2 The Lasso

p개의 모든 예측변수를 포함한다는 점은 ridge의 큰 단점이다. 다른 불필요한 변수에 대한 계수 추정치를 정확히 0으로 만드는 것이 아니기 때문에 여전히 이러한 변수들이 모델에 포함되어 있다. ridge의 penalty인 $\lambda \sum \beta_j^2$는 모든 계수 추정치를 정확히 0으로 보내지는 않을 것이다.($\lambda = \infty$가 아닌 이상) 이것은 prdeiction accuracy 문제에 관해서는 상관 없지만 model interpretation에 관해서는 문제가 될 수 있다.

lasso는 이러한 ridge의 단점을 보완하는 대안책이다. lasso 계수인 $\hat{\beta_\lambda}^L$은 아래 식을 최소화한다.

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} | \beta_j | = Q(\beta) + \lambda \sum_{j=1}^{p} | \beta_j | \tag{6.7}$$

6.5와 6.7 을 비교하면, lasso와 ridge는 비슷한 공식을 가지고 있음을 알 수 있다. 한 가지 차이점은 β_j^2 term이다. lasso는 ℓ_2대신에 ℓ_1을 사용한다. 백터 β에 대한 ℓ_1 norm은 $\|\beta\|_1 = \sum | \beta_j |$이다. ℓ_2 norm은 $\|\beta\|_2 = \sum \beta_j^2$ 이다.

lasso에서는 $\ell_1$penalty는 tuning parameter인 λ에 따라서 어떠한 계수가 정확히 0에 가깝게 만드는 효과가 있다. 따라서 best subset selection과 비슷하게, lasso는 variable selection을 하는 것이다. 따라서 variable selection에 의해서 계수의 수가 적어지기 때문에 lasso로

만들어진 모델이 ridge로 만들어진 모델보다 해석하기는 훨씬 쉽다.

Variable Selection Property of the Lasso

Another Formulation for Ridge Regression and the Lasso

$$minimize_{\beta}\left\{\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2\right\} \text{ subject to } \sum_{j=1}^{p}\mid\beta_j\mid\leq s \tag{6.8}$$

$$minimize_{\beta}\left\{\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2\right\} \text{ subject to } \sum_{j=1}^{p}\beta_j^2\leq s \tag{6.9}$$

다시 말해서 모든 λ의 값에 대해서 6.7 과 6.8이 동일한 lasso 계수 추정치를 내는 s가 존재하고 6.5와 6.9가 동일한 ridge 계수 추정치를 내는 s가 있다는 것이다.
6.8을 다음과 같이 생각할 수도 있다. 다시 말해, $\sum_{j=1}^{p}\mid\beta_j\mid$가 얼마나 큰지에 대해서 s라는 값으로 budget의 제한이 있는 것이다. s가 엄청 크면 이 budget은 제한적이지 않아 그에 따른 계수 추정치도 매우 클 수 있다. 사실 s가 충분히 커서 최소자승 추정치가이 budget안에 포함된다면 6.8은 least squares solution과 같은 결과를 낼 것이다. figure 6.7을 보자. 예측변수가 두 개인 간단한 상황이다. 왼쪽 그림에서 뾰족한 마름모를 확인할 수 있는데 이는 lasso의 제약식을 표현한 것이다$\left(\sum_{j=1}^{p}\mid\beta_j\mid\leq s\right)$ 가운데의 검정색 점과 $\widehat{\beta}$라고 표시된 것은 $Q(\beta)$를 최소화하는 최소자승 추정치를 의미한다. 그리고 그 주변의 빨강색 타원은 다양한 값의 $Q(\beta)$을 의미한다. 만약 s의 값이 커져서 최소자승 추정치인 $\widehat{\beta}$을 포함해버린다면, lasso의 해는 최소자승 추정치와 동일하게 될 것이다(이렇게 s가 큰 상황은 $\lambda=0$인 상황과 동일하다)
그와 반면에, s가 작다면lasso 추정치는 budget을 위반하는 것을 피하기 위해 작아야만 할 것이다. 6.9도 마찬가지로 생각하면 된다.

The Variable Selection Property of the Lasso

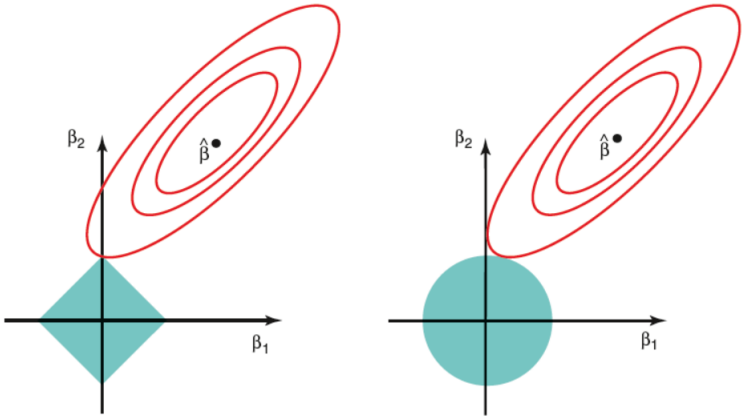


FIGURE 6.7. *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions,* $|\beta_1| + |\beta_2| \leq s$ *and* $\beta_1^2 + \beta_2^2 \leq s$, *while the red ellipses are the contours of the RSS.*

왜 lasso가 ridge와는 다르게 계수 추정치에서 정확히 0과 똑같은 결과를 낼까?
ridge는 뾰족한 점 없이 동그란 제약 조건을 갖기 때문에 타원과 이 제약 지역간의 교점은 보통 축에서 일어나지는 않을 것이고 따라서 ridge 계수 추정지는 0과 가까울 수는 있지만 정확히 0이기는 힘들다. 하지만 lasso는 축에 corners을 가지고 있기 때문에 타원이 이 축에서 만날 것이고 이러한 상황에서 계수 중 하나는 0이 될 것이다. 고차원에서는 많은 계수 추정치들이 동시에 0이 될 것이다.

Comparing the Lasso and Ridge Regression

lasso가 몇몇개의 predictors을 포함하는 더 간단하고 해석력있는 모델을 만든다는 점에서 ridge에 비해 더 큰 장점이 있다는 것은 자명하다. 하지만 어떠한 방법이 더 나은 prediction accuracy을 가질까?

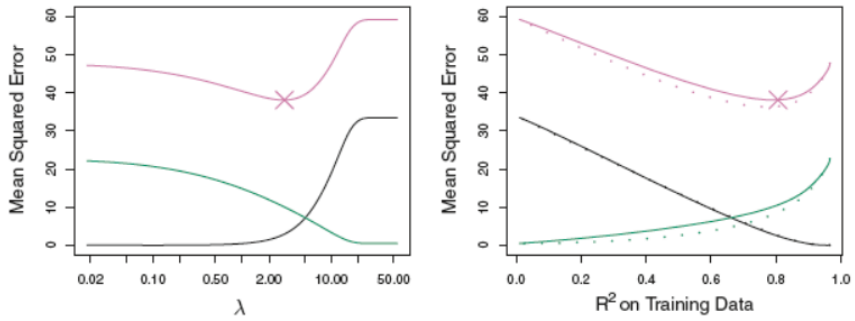


FIGURE 6.8. Left: *Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso on a simulated data set.* Right: *Comparison of squared bias, variance and test MSE between lasso (solid) and ridge (dotted). Both are plotted against their* R^2 *on the training data, as a common form of indexing. The crosses in both plots indicate the lasso model for which the MSE is smallest.*

그림 6.8은 lasso의 variance, bias의 제곱, 그리고 test MSE를 보여준다. 분명히, 왼쪽 그림에서 λ가 커질수록 variance는 감소하고 bias는 증가한다는 점에서 lasso는 ridge와 비슷한 성향을 가진다. 오른쪽 그림에서 점은 ridge regression 적합을 나타낸다. 오른쪽 그림에서 ridge와 lasso는 bias에서는 거의 비슷하지만 ridge가 lasso에 비해 variance가 살짝 낮고 그에 따라서 최소 MSE도 lasso보다 살짝 낮다. 왼쪽 그림의 x축은 λ이고 오른쪽 그림의 x축은 R^2이다. 이렇게 훈련 데이터에 대한 R^2로 그림을 그리는 것도 모델을 비교할 때 많이 사용한다.

하지만 그림 6.8의 데이터는 모든 45개의 변수가 반응변수와 연관되어 있는 경우이다. 다시 말해서, 모든 45개의 변수의 실제 계수 값(parameter)가 0이 아닌 상황에서 데이터를 임의로 생성한 것이다. lasso는 암묵적으로 몇 개의 계수가 완전시 0과 똑같다고 가정한다. 결과적으로 ridge가 lasso보다 이 상황에서는 더 좋은 성능을 보이는 것이 자명하다.

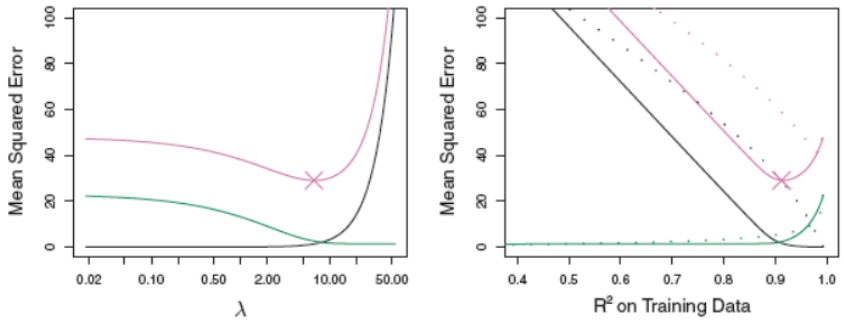


FIGURE 6.9. Left: *Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso. The simulated data is similar to that in Figure 6.8, except that now only two predictors are related to the response.* Right: *Comparison of squared bias, variance and test MSE between lasso (solid) and ridge (dotted). Both are plotted against their R^2 on the training data, as a common form of indexing. The crosses in both plots indicate the lasso model for which the MSE is smallest.*

그림 6.9에서는 반응변수를 단 두 개의 변수로만 생성하여 나머지 43개의 변수는 쓸모 없는 변수로 설정했다. 이제 lasso가 bias, variance, MSE에 관해서 모두 ridge의 그래프보다 아래에 있음을, 즉 ridge보다 성능이 훨씬 좋음을 확인할 수 있다.
위의 두 예시에서 알 수 있듯이, ridge와 lasso가 어느 경우에서나 다른 것보다 좋다고 보장할 수는 없다. 일반적으로 lasso는 적은 변수들이 중요한 변수이고 나머지 계수는 매우 작거나 0과 같을 때 좋은 성능을 낼 수 있다. ridge는 반응변수가 많은 예측변수의 함수일때, 모든 변수가 대략적으로 비슷한 크기일 때 더 좋은 성능을 낸다. 하지만 반응변수와 연관되어 있는 예측변수의 수는 실제 생활에서 절대로 먼저 알려져 있지 않는다. cross-validation같은 기술들이 어떠한 방법이 나은지를 위해 사용될 수 있다.

The Elastic Net

여태까지 살펴보았던 ridge와 lasso를 아주 간략하게 정리하면 다음과 같다. lasso는 variable selection을 통해서 불필요한 변수에의 계수를 0으로 만드므로 불필요한 변수가 많이 포함되어 있는 상황에서 좋은 성능을 보이고 반면에 모든 계수가 전반적으로 반응변수와 좋은 관계를 보일 때에는 계수를 0으로 만들어버리지 않는 ridge을 쓰는 것이 좋다. 하지만 데이터를 보고 어떻게 예측변수와 반응변수간의 관계를 알 수 있을까? 그러한 관계를 사전에 모르기 때문에 ridge, lasso 중에 어느 것을 써야할지 고민이 생길 것이다. elastic net은 ridge와 lasso의 절충 모델이다. 즉, elastic net은 아래와 같은 식을 최소화하고자 한다.

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

penalty항이 ridge와 lasso를 모두 포함한 것을 확인할 수 있다. 따라서 ridge와 lasso의 단점은 보완하고 둘의 장점을 잘 담아낸 모델이다.

6.2.3 Selecting the Tuning Parameter

cross-validation을 통해서 tuning parameter인 λ을 고를 수 있다. 후보 λ 값들에 대해서 k-fold cross validation을 시행하는 과정을 생각해보자. chapter 5에서 살펴보았듯이, 우선, 데이터를 훈련 데이터와 테스트 데이터로 나누고, 훈련 데이터를 k개의 겹으로 나눈다. 그리고 k-1개의 겹으로 어떤 λ 값에 대해서 모델을 생성한 후, 나머지 1개의 겹으로 해당 λ의 성능을 파악한다. 이러한 과정을 동일한 λ에 대해서 k번을 한 후, 여기서 나온 k개의 cross-validation error을 다른 모든 λ 값에 대해서 비교한다. 그리고 가장 좋은 결과를 내는 λ을 골라서, 다시 처음에 나눈 훈련 데이터로 모델을 생성한다.

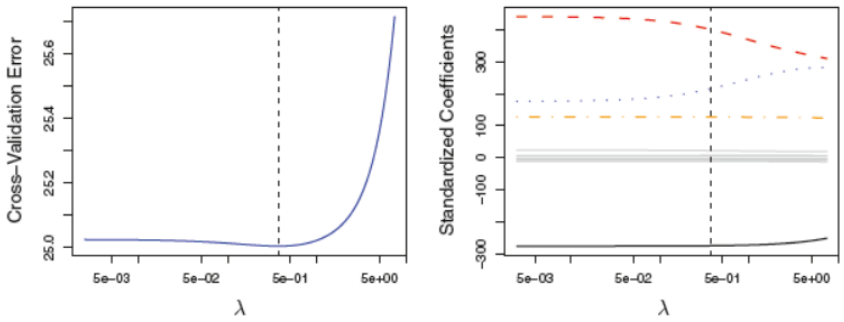


FIGURE 6.12. Left: *Cross-validation errors that result from applying ridge regression to the* `Credit` *data set with various value of* λ. Right: *The coefficient estimates as a function of* λ. *The vertical dashed lines indicate the value of* λ *selected by cross-validation.*

figure 6.12는 leave-one-out cross-validation의 시행 결과에 따른 λ의 선택을 보여준다. 여기서 수직 점선은 선택된 λ을 의미하고 이 경우에서는 그 값이 상대적으로 작은데 이것은 least squares solution에 대해서 작은 양의 shrinkage만 함을 의미한다. 그리고 움푹 패인 부분도 확실하지 않아서 이러한 경우에는 그냥 least squares solution을 쓰는 것이 나을 수도 있다.

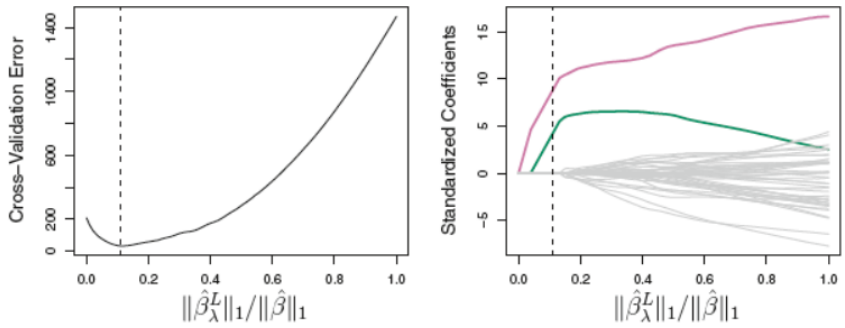


FIGURE 6.13. Left: *Ten-fold cross-validation MSE for the lasso, applied to the sparse simulated data set from Figure 6.9.* Right: *The corresponding lasso coefficient estimates are displayed. The vertical dashed lines indicate the lasso fit for which the cross-validation error is smallest.*

figure 6.13은 figure 6.9 데이터에 대해서 lasso를 ten-fold- cross-validation으로 시행한 결과를 보여준다. 왼쪽 그림은 CV error이고 오른쪽 그림은 계수 추정치를 나타낸다. 수직 점선은 CV error가 최소일 때이다. 오른쪽 그림의 두 색의 선은 반응변수와 연관된 두 개의 예측변수를, 회색 선들은 관련되지 않은 변수(signal와 noise라고 각각 부른다.)들을 나타낸다. lasso가 두 signal 예측 변수에 대해 맞게 더 큰 계수 추정치를 줬을 뿐만 아니라 CV error도 최소화 했고 바로 이 점이 signal 예측 변수만 0이 아닌 지점이다. 나머지 noise 변수는 모두 0이다. 따라서 lasso과 cross-validation 세트메뉴는 두 개의 signal 변수를 아주 잘 맞추었다! 하지만 least squares는 오른쪽 그래프에서 완전 오른쪽인데(x축이 1이라는 것은 분모 분자가 같다는 것이고 이것은 least squares method를 시행했을 때이다.) 이때에는 두 개의 signal 변수 중 하나만 높은 값을 주었고 noise변수도 많이 있다.

6.3 Dimension Reduction Methods

이 챕터에서 여태까지 논의해온 방법은 원래의 변수의 subset을 사용하거나 그들의 계수를 shrink함으로써 variance을 조절했다. 이러한 모든 방법은 원래의 변수, 즉, $X_1, X_2, \cdots, X_p$ 을 이용하여 정의된다. 이제 변수를 transform하여 이러한 변환된 변수를 이용하여 least squares model을 적합시키는 방법을 배운다. 이러한 방법을 dimension reduction(차원축소)를 통한 회귀분석이라고 부른다. 차원 축소의 개념부터 살펴보자.

$Z_1, Z_2, \cdots, Z_M$ 을 $M < p$ 우리의 원래 p개의 변수의 linear combinations를 나타낸다고 하자. 다시 말해

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j \tag{6.16}$$

인데 여기서 $\phi_{1m}, \phi_{2m}, \cdots, \phi_{pm}$ 은 상수고 $m = 1, \cdots, M$ 이다. 그러면 다음과 같은 linear

regression LSE를 이용하여 적합할 수 있다.

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \epsilon_i, \text{ for } i = 1, \cdots n \tag{6.17}$$

dimension reduction이라는 용어는 이러한 접근법이 $p+1$개의 계수들, 즉, $\beta_0, \beta_1, \cdots, \beta_p$를 추정하는 것에서 좀 더 간단한 문제인 $M+1$개의 계수를 추정하는 문제, 즉, $\theta_0, \theta_1, \cdots, \theta_M$(여기서 $M < p$) 로 바뀌었다는 점에서 유래한다. 다시 말해서 차원이 $p+1$에서 $M+1$로 줄어든 것이다.

6.16에서 다음을 주목하자.

$$\sum_{m=1}^{M} \theta_m z_{im} = \sum_{m=1}^{M} \theta_m \sum_{j=1}^{p} \phi_{jm} x_{ij} = \sum_{j=1}^{p} \sum_{m=1}^{M} \theta_m \phi_{jm} x_{ij} = \sum_{j=1}^{p} \beta_j x_{ij}$$

$$where\ \beta_j = \sum_{m=1}^{M} \theta_m \phi_{jm} \tag{6.18}$$

따라서 6.17은 원래의 선형 회귀 모델의 특별한 케이스라고 생각할 수 있다. 차원 축소 기법은 β_j 계수 들을 constrain하는 역할을 하는데, 이는 그 계수들이 6.18과 같은 형태를 취하기 때문이다. 이러한 제약은 계수가 추정하는 것에 대한 잠재적인 bias의 가능성이 있다. 어떤 제약을 준다는 것은 그 제약이 올바르지 못할 때에 실제 모수와 동떨어진 추정치를 만들어낼 수 있기 때문이다. 하지만 MSE가 bias으로만 이루어지지 않는 다는 것을 다시 생각해보자. p가 n에 비해서 상대적으로 클 때, $M << p$인 M을 고르는 것은 적합된 계수의 variance을 상당히 줄여준다.

6.3.1 Principal Components Regression

여기서 PCA를 회귀를 위한 차원 축소 기술로써 그것의 사용법에 대해서 얘기해본다.

An Overview of Principal Components

PCA는 $n \times p$의 데이터 행렬 $\mathbf{X}$의 차원을 축소하는 기법이다.

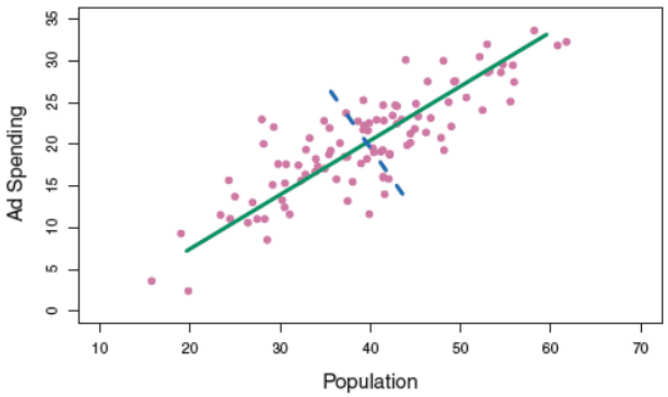
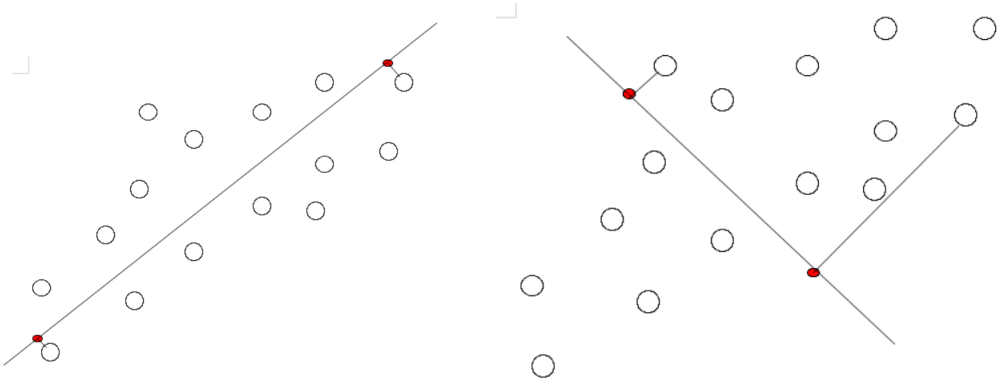


FIGURE 6.14. *The population size (pop) and ad spending (ad) for 100 different cities are shown as purple circles. The green solid line indicates the first principal component, and the blue dashed line indicates the second principal component.*

The first principal component direction of the data는 관측치 중 가장 많이 변하는 것이다. 예를 들어 figure 6.14에서 초록색 선은 데이터의 first principal component direction을 나타낸다. 이 선이 데이터에서 가장 큰 variability을 가지는 선인데, 다시 말해서 100개의 관측치를 이 선으로 project해서 나오는 resulting projected observations는 가장 큰 가능한 variance을 가질 것이다. 다른 어떤 선에 project을 하더라도 이 선보다 작은 variance가 나올 것이다(여기서 line에 project한다는 것은 그 선과의 최소 거리를 찾는다는 것임).



위 그림을 살펴보자. 왼쪽 그림과 오른쪽 그림은 동일한 점들이다. 왼쪽 그림에서 어떤 선을 그렸는데, 그선으로 점들을 수직으로 이동(projection)시켜보자. 그러면 원래 2차원이었던 점들이 차원이 축소되어 1차원 위에 위치하게 된다. 오른쪽 그림도 그어진 선으로 동일하게 점들을 수직으로 이동시켜보자. 선 위에 있는 점들의 모습을 생각하면, 왼쪽 그림에서 점들이 더 퍼져있음을 알 수 있다. 즉, 더 퍼져있다는 것을 통계적으로 얘기하면 분산이 크다는

것이다. 다시 말하면, 원래 점이 퍼져있는 정도(정보)를 왼쪽 그림이 오른쪽 그림보다 더 잘 보존한다고 말할 수 있다. PCA는 이렇게 분산을 최대화(=정보량을 최대로 보존하는) 하게 projection을 진행하여 차원을 축소한다.

first principal component은 figure 6.14에 그래프 상으로 나와있다. 이것을 식으로 나타내면 다음과 같다.

$$Z_1 = 0.839 \times (pop - \overline{pop}) + 0.544 \times (ad - \overline{ad}) \tag{6.19}$$

여기서 $\phi_{11} = 0.839$와 $\phi_{21} = 0.544$은 principal component loadings인데 이것은 위에서 언급된 direction을 정의한다. 6.19에서 $\overline{pop}$은 데이터 셋의 모든 pop values의 평균을 나타내고 $\overline{ad}$도 마찬가지이다. 이러한 아이디어는 $\phi_{11}^2 + \phi_{12}^2 = 1$을 만족하는 모든 pop과 ad의 linear combinations 중에서 6.19가 가장 큰 variance을 가짐을 의미한다. 즉, $Var(Z_1)$이 최대가 됨을 의미한다.

$n = 100$이기 때문에 pop과 ad는 길이가 100인 백터이고 6.19의 Z_1도 그러하다. 예를 들어,

$$z_{i1} = 0.839 \times (pop - \overline{pop}) + 0.544 \times (ad - \overline{ad}) \tag{6.20}$$

이고 $z_{11}, \cdots, z_{n1}$은 principal component scores라고 부르며 figure 6.15 오른쪽에서 볼 수 있다.

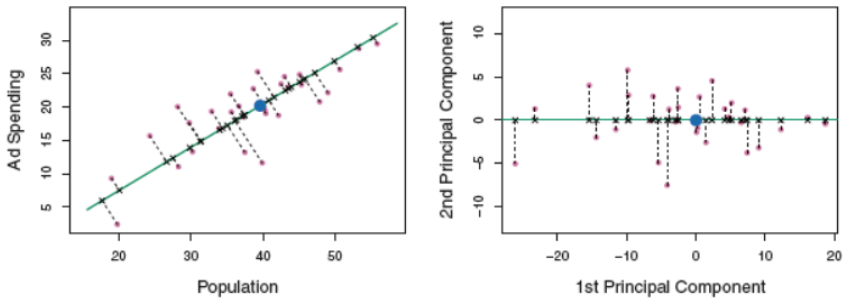


FIGURE 6.15. *A subset of the advertising data. The mean* pop *and* ad *budgets are indicated with a blue circle.* Left: *The first principal component direction is shown in green. It is the dimension along which the data vary the most, and it also defines the line that is closest to all n of the observations. The distances from each observation to the principal component are represented using the black dashed line segments. The blue dot represents* ($\overline{\text{pop}}, \overline{\text{ad}}$). Right: *The left-hand panel has been rotated so that the first principal component direction coincides with the x-axis.*

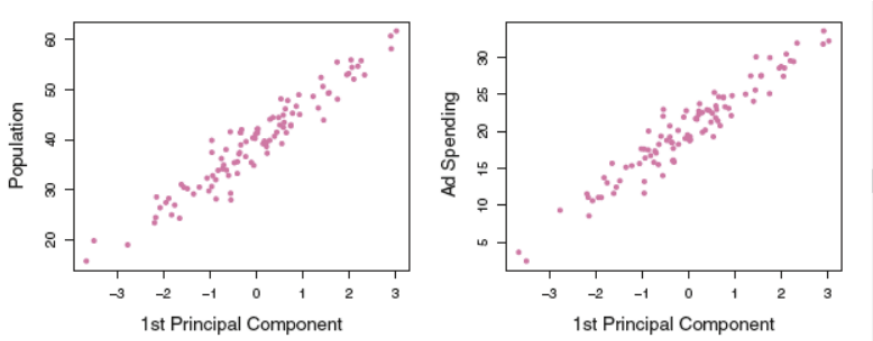


FIGURE 6.16. *Plots of the first principal component scores z_{i1} versus* pop *and* ad*. The relationships are strong.*

figure 6.16은 first principal component와 두 변수간의 관계를 보여주는데 선형의 관계가 강함을 알 수 있고 이것은 다른 말로 first principal component가 pop과 ad에 담겨져 있는 대부분의 정보를 잘 포함한다는 것을 의미한다.

여태까지 first principal component에 주목해왔다. 일반적으로는 p개의 principal components까지 만들 수 있다. Second principal component인 Z_2는 Z_1와 관련이 없는 변수의 linear combinations이다. Second는 figure 6.14에 파란색 선으로 나타나 있다. Z_1과 Z_2가0의 상관계수를 가진다는 것은 그 direction이 first principal component와 반드시 수직 또는 orthogonal 하다는 것과 동일하다. Second는 다음과 같다.

$$Z_2 = 0.544 \times (pop - \overline{pop}) - 0.839 \times (ad - \overline{ad})$$

광고 데이터가 두 개의 예측변수만을 포함하기 때문에, first principal component의 두 개의 변수가 pop과 ad의 모든 정보를 포함한다. 그에 따라서 second을 만들어봤자 pop과 ad와 별로 관련이 없는 것이 만들어진다. figure 6.17 참조.

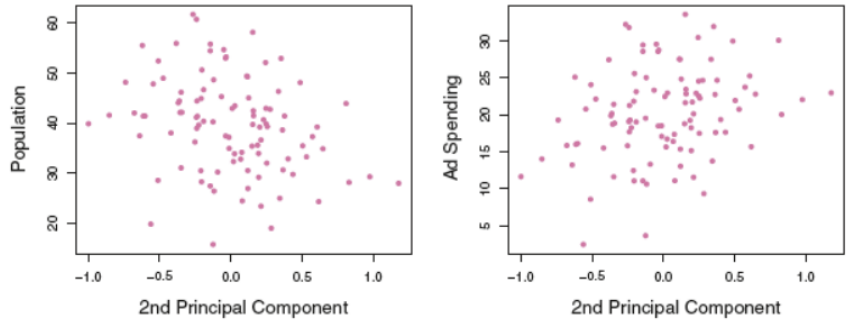


FIGURE 6.17. *Plots of the second principal component scores z_{i2} versus* pop *and* ad*. The relationships are weak.*

2차원 데이터에서는 많아야 두 개의 principal components을 만들 수 있다. 하지만 변수가 더 여러개면 추가적인 components을 만들 수 있다. 그들은 연속적으로 이 전의 components와 관련이 없다는 제약 아래 연속적으로 variance을 maximize할 것이다.
사실 R이나 python 등 프로그래밍을 통해서 PCA를 쉽게 할 수 있지만 그래도 어떻게 principal component direction이 생성되는지 한번 쯤은 살펴보고 갈 필요가 있다. 아래의 증명을 통해서 사실, principal component direction은 사실 eigenvectors라는 것을 밝힐 수 있다.

$\sum = VDV^{'}$을 $X_1, \cdots, X_p$의 covariance matrix라고 두고 이를 eigen decomposition(정사각 행렬에 대해서 할 수 있는 분해)을 통해 $VDV^{'}$로 다르게 표현했다. 여기서 $V = [v_1, \cdots, v_p], D = Diag\{\lambda_1, \cdots, \lambda_p\}\ with\ \lambda_1 > \cdots > \lambda_p > 0$. 여기서 $\sum = VDV^{'} = \sum_{i=1}^{p} v_i \lambda_i v_i^{'}$이다. $X_1, \cdots, X_p$의 linear combinations인 ℓX의 분산은 $Var(\ell^{'} X) = \ell^{'} \sum \ell$이다. 주의할 점은, 벡터 ℓ은 normalized 벡터라는 것이다. 즉, $\ell^{'} \ell = 1$이다. 여기서 eigen vector인 $v_1, \cdots, v_p$은 Euclidean Space에서의 orthogonal basis이다. 따라서 벡터 ℓ을 다음과 같이 나타낼 수 있다. $\ell = c_1 v_1 + \cdots + c_p v_p = [v_1, \cdots, v_p] \begin{bmatrix} c_1 \\ \cdots \\ c_p \end{bmatrix} \equiv Vc$ (basis들의 linear combination)

$$
\begin{aligned}
Var(\ell^{'} X) &= \ell^{'} \sum \ell \\
&= \ell^{'} VDV^{'} \ell \\
&= c^{'} V^{'} VDV^{'} Vc \\
&= c^{'} Dc \\
&= \sum_{k=1}^{p} \lambda_k c_k^2
\end{aligned}
$$

한편, $\ell^{'} \ell = c^{'} V^{'} Vc = c^{'} c = \sum_{k=1}^{p} c_k^2 = 1$이다. 즉, $Var(\ell^{'} X)$은 eigen values인 λ_k에 대한 weighted average라고 볼 수 있다. (c_k^2가 가중치인 형태) 그런데 λ_k은 앞서 값의 크기에 따라 배열을 했으므로, λ_1이 가장 큰 값이다. 따라서 λ_k에 대한 weighted average인 $Var(\ell^{'} X)$을 최대로 하려면 가장 큰 값에 모든 가중치를 두면 된다. 다시 말하면, $c_1^2 = 1, c_2^2 = \cdots = c_p^2 = 0$으로 설정하는 것이다. 이를 이용하면 $\ell = c_1 v_1 + \cdots + c_p v_p = v_1\ or\ -v_1$이다(부호는 어차피 동일한 공간에 있기 때문에 문제가 되지 않는다) 즉, first principal direction이 바로 첫 번째 eigenvector인 것이다. 이를 $\ell_1 = v_1$이라 두자. k+1 th principal direction 부터는 그 이전의 principal direction과 orthogonal해야 하므로 아래와 같은 제약이 생긴다. $v_j^{'} \ell = v_j^{'}(c_1 v_1 + \cdots + c_p v_p) = c_j v_j^{'} v_j = c_j = 0\ for\ j = 1, \cdots, k$ 이러한 제약 아래 $\sum_{k=1}^{p} \lambda_k c_k^2$

을 최대로 하기 위해서는 c_{k+1}에 모든 가중치를 두면 된다. 따라서 이때에 $\ell_{k+1} = v_{k+1}$이다.

The Principal Components Regression Approach

PCR은 M개의 principal components, $Z_1, \cdots, Z_M$를 만든 후 얘네들을 least squares을 적합하는 선형 회귀 모델에서 예측 변수로써 사용한다. 핵심은 종종 적은 수의 principal components로도 데이터 대부분의 변동성과 반응변수와의 관계를 설명하기에 충분하다는 것이다. 다시 말해서 $X_1, \cdots, X_p$가 보여주는 가장 큰 variation인 direction이 Y와 가장 관련된 direction이라는 가정을 한다. 이러한 가정은 사실임이 보장되지는 않지만 좋은 결과를 내기 위한 충분히 합리적인 근사임을 알 수 있다.
만약 PCR의 가정이 유효하다면, $Z_1, \cdots, Z_M$에 least squares 모델을 적합하는 것은 $X_1, \cdots, X_p$에 least squares 모델을 적합하는 것보다 더 나은 결과를 가져오는데 이는 거의 대부분의 정보가 principal components에 포함되어 있기 때문이고 p개 전체를 사용하는 것이 아니라 $M << p$인 M개의 계수를 사용함으로써 overfitting을 완화할 수 있기 때문이다.

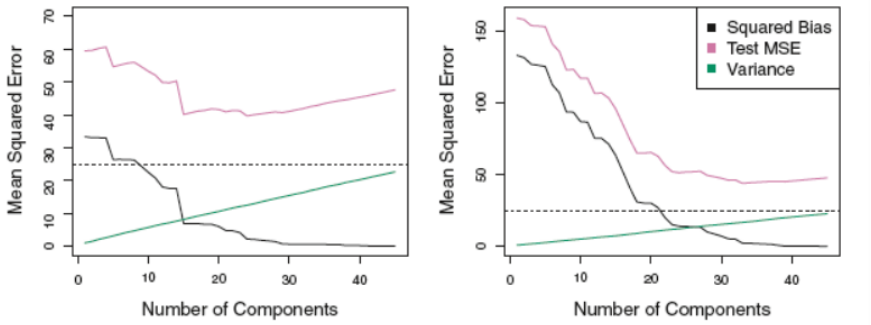


FIGURE 6.18. *PCR was applied to two simulated data sets.* Left: *Simulated data from Figure 6.8.* Right: *Simulated data from Figure 6.9.*

figure 6.18은 figure 6.8, 6.9의 데이터에 대한 PCR 적합이다. 첫 번째 데이터에서 반응변수는 모든 변수를 사용한 함수였고 두 번째 데이터에서 반응변수는 두 개의 변수만 사용한 함수였다. 위 그래프는 M(regression model에서 예측변수로 사용된 principal components의 갯수)에 대한 함수이다.
왼쪽 그림을 보면 M이 커질 수록, bias는 낮아지지만 variance는 증가한다. 이것으로 인해서 Test MSE가 U자 모양이다. 그리고 만약 $M = P = 45$라면 PCR은 least squares 적합과 동일한 결과를 낸다. 적절한 M의 선택이 least squares보다 훨씬 좋은 성능을 낼 수 있음을 보여준다. 하지만 ridge와 lasso에 실험을 한 결과 PCR은 원래의 shrinkage methods만큼 좋은 성능을 내는 것은 아님을 알 수 있다.

figure 6.18에서 왜 PCR이 더 나쁜 성능을 보일까?? 그것은 아마도 적절한 모델을 만들기 위해 많은 principal components이 필요하기 때문일 것이다. 반면에 처음의 적은 principal components으로써 충분하게 예측변수의 변동 뿐만 아니라 반응벼눗와의 관계도 잡아낼 수 있다면 PCR은 좋은 성능을 낼 것이다.

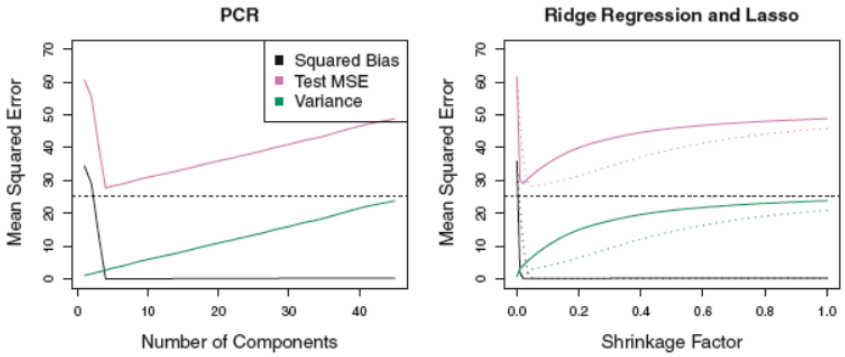


FIGURE 6.19. *PCR, ridge regression, and the lasso were applied to a simulated data set in which the first five principal components of X contain all the information about the response Y. In each panel, the irreducible error $Var(\epsilon)$ is shown as a horizontal dashed line.* Left: *Results for PCR.* Right: *Results for lasso (solid) and ridge regression (dotted). The x-axis displays the shrinkage factor of the coefficient estimates, defined as the ℓ_2 norm of the shrunken coefficient estimates divided by the ℓ_2 norm of the least squares estimate.*

figure 6.19에서 왼쪽 그림은 PCR에 좀더 호환적인 데이터셋에 대한 결과를 보여준다. 여기서 반응변수는 첫 다섯개의 principal components에 전적으로 의존하게 생성되었다. 이제 bias는 M이 증가함에 따라 급격하게 0으로 간다. 그리고 Test MSE도 $M = 5$일 때, 분명한 최소값을 가진다. figure 6.19에서 오른쪽 그림은 이 데이터에 lasso와 ridge을 적용한 결과를 보여준다. 이 세가지 방법은 least squares에 비해서 더 나은 결과를 보여주지만 PCR과 ridge가 lasso보다 훨씬 더 성능이 좋다.

여기서 주목해야할 점은 PCR이 $M < p$인 M개의 예측변수를 사용하는 간단한 방법이라고 할지라도 이것은 feature selection method가 아니라는 것이다. 왜냐하면 M개의 principal components가 기존의 features의 linear combination이기 때문이다. 따라서 원래의 변수들에서 소수의 변수만 골라서 만들어진 모델이 아니라는 점에서 PCR은 lasso보다는 ridge와 더 밀접하게 연관이 되어있다. ridge 회귀가 PCR의 연속 버전이라고 생각할 수도 있다!

PCR에서는 principal components의 갯수인 M은 cross-validation으로 결정된다. cross-validation을 통해서 얻어진 가장 낮은 CV error을 가지는 M을 택하는 것이다.

PCR을 할 때, 일반적으로 principal components을 생성하기 이전에 각각의 변수를 6.6을 이용하여 standardizing을 하는 것이 좋다. 이러한 표준화는 모든 변수가 확실하게 동일한 scale에 있게 해준다. 표준화를 하지 않는다면 높은 variance을 가지는 변수가 principal

components에서 주요한 역할을 할 것이고 이것이 결국 PCR모델에 영향을 미칠 것이다. 하지만 모든 유닛이 똑같은 방식으로 측정이 된다면(예를 들어 모두 kg, m로 측정이 된다면) 표준화를 하지 않아도 된다.

6.3.2 Partial Least Squares

PCR은 예측 변수 $X_1, \cdots, X_p$를 가장 잘 표현하는 linear combinations 또는 directions을 찾는 과정이다. 이러한 과정은 unsupervised(비지도) 방법인데 이는 반응변수 Y가 principal component directions을 결정하는데 사용되지 않기 때문이다. 그에 따라 PCR은 한 결점을 가지고 있다. 예측변수를 가장 잘 설명하는 directions이 반응변수를 설명하는 가장 좋은 directions이라는 보장이 없다. 애초에 principal component direction을 만들 때 반응 변수를 고려하지 않기 때문이다.

이제 partial least squares(PLS)을 소개하는데 얘는 PCR에 대한 대안으로 나온 supervised(지도) 학습이다. PCR 처럼 PLS은 차원축소 방법인데 처음에 원래 변수의 linear combination 조합인 $Z_1, \cdots, Z_M$을 찾고 이 M개의 새로운 변수를 이용하여 least squares을 통해 선형 모델을 적합하는 방법이다. 하지만 PCR과는 다르게 PLS는 이러한 새로운 features를 지도학습 방법으로 알아낸다. 다시 말해서, PLS는 이전의 변수와 비슷할 뿐만 아니라 반응변수와도 연관이 되어 있는 새로운 변수를 알아내기 위해 반응변수 Y를 사용한다.

이제 첫 PLS가 어떻게 계산되는지 살펴보자. p개의 변수들을 표준화 후, PLS는 6.16의 ϕ_{j1}을 simple linear regression of Y onto X_j의 계수와 동일하게 설정함으로써 첫 번째 direction인 Z_1을 계산한다. (단순 선형 회귀 계수는 Y와 X_j의 correlation에 비례[proportional]한다.) 따라서 $Z_1 = \sum_{j=1}^{p} \phi_{j1} X_j$을 계산하는데, PLS는 반응변수와 가장 관련이 강한 변수에 가장 큰 비중을 둔다.

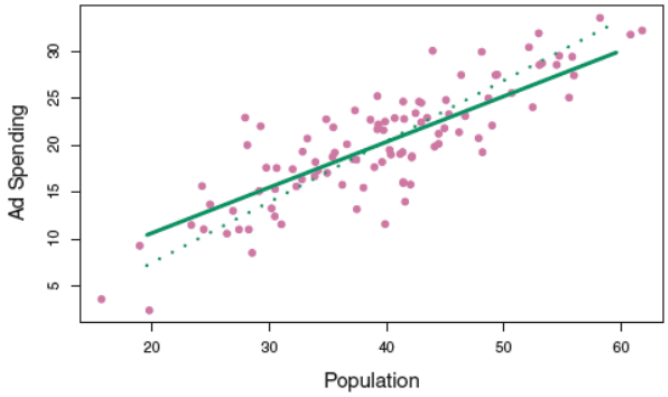


FIGURE 6.21. *For the advertising data, the first PLS direction (solid line) and first PCR direction (dotted line) are shown.*

figure 6.21은 sales을 반응변수로 하고 두 개의 예측변수에 대한 PLS 예시이다. 초록색 실선은 first PLS direction을 의미하는 반면 점선은 first principal component direction을 의미한다. PLS는 PCA에 비해서 ad dimension에서 더 적은 변화를 가지는 direction을 선택했다. 즉, pop가 한 단위 증가할 때 ad가 덜 증가하는 직선이다. 이것은 pop이 ad보다 반응변수와 더 깊게 연관되어 있음을 의미한다. PLS는 PCA만큼 예측 변수들을 가깝게 적합시키지는 않지만 반응변수를 설명하는 측면에서는 강점을 가진다.
Second PLS direction을 확인하기 위해서 각각의 변수를 Z_1에 대해 regressing함으로써 각각의 변수를 Z_1에 대해 adust한 후, residuals을 구한다. 이러한 잔차는 first PLS direction에 의해서 설명되지 않는 나머지 정보로 해석될 수 있다. 그리고 orthogonalized 데이터(수직의 데이터라는 말은 서로 연관이 없다, 독립이다, Z_1로 설명하고 남은 부분은 Z_1로 설명한 부분과 독립이라는 뜻을 내포)를 이용해서 Z_1을 원래의 데이터에 기반해서 계산했던 것과 동일한 방식으로 Z_2을 계산한다. 이러한 과정을 M번 반복해서 여러개의 PLS components $Z_1, \cdots, Z_M$을 만든다. 그리고 마지막으로 least squares을 이용해서 Y을 예측하기 위해 $Z_1, \cdots, Z_M$을 이용하여 linear model을 적합시킨다.
PCR과 마찬가지로 PLS에서 사용되는 M도 CV에 의해서 결정되는 tuning parameter이다. 또한 일반적으로 PLS를 하기 이전에 예측변수들을 표준화한다.

6.4 Considerations in High Dimensions

6.4.1 High-Dimensional Data

20세기에서는 마케팅, 의학 분야 등 변수(feature)의 갯수가 매우 많은 데이터가 다반수다. 하지만 비용, sample availability 등으로 인해서 관측치의 갯수인 n은 p에 비해서 제한적이다. 이렇게 변수가 관측치의 갯수보다 더 많은 데이터를 high-dimensional하다고 한다. least squares linear regression과 같은 고전적인 접근법은 이러한 상황에 적절하지 않다. 이제부터 할 얘기는 고차원 데이터에 뿐만 아니라 p가 n보다 살짝 작은 경우에도 적용될 수 있으니 지도학습(supervised learning)을 시행할 때 주의하자!

6.4.2 What Goes Wrong in High Dimensions?

고차원 데이터에 대해서 회귀 분석이나 분류를 할 때 좀 더 각별한 주의가 왜 필요한지 설명하기 위해 고차원 데이터에 맞게 만들어지지 않은 통계적 기법을 적용했을 때의 문제점에 대해서 살펴보자. 이것을 위해서 least squares regression을 테스트할건데, 로지스틱 회귀나 다른 고전적인 기법들에 대해서 동일하게 적용된다.
고차원 데이터에 대해서 least squares는 시행될 수 없다. 이유는 간단하다: 설명변수와 반응변수 간에 진정한 관계가 있든 없든, least squares는 데이터에 완벽하게 적합되는, 잔차가 0인 계수 추정치를 만들 수 있기 때문이다.

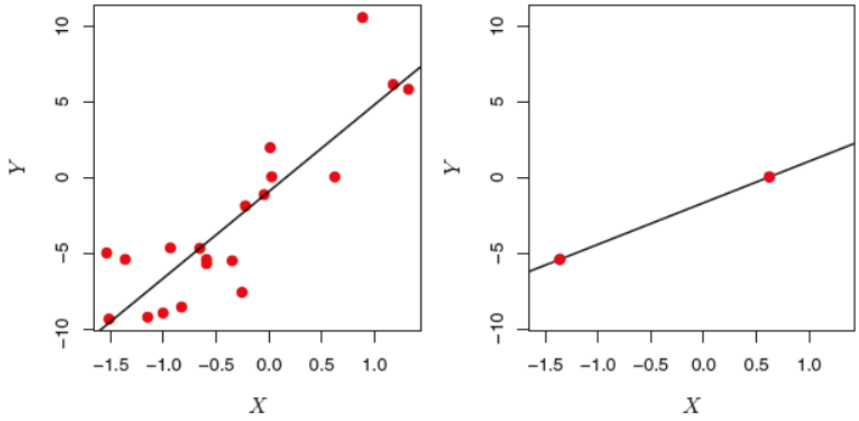


FIGURE 6.22. Left: *Least squares regression in the low-dimensional setting.* Right: *Least squares regression with $n = 2$ observations and two parameters to be estimated (an intercept and a coefficient).*

figure 6.22에는 한개의 변수를 가지는 선형 모델인데 왼쪽 그림은 20개의 관측치, 오른쪽은 2개의 관측치가 표현되어 있다. 왼쪽 그림은 least squares regression line이 데이터에 완벽하게 적합되지 않는다. 대신, 회귀 선은 20개의 관측치를 최대한 근사시키려고 한다. 하지만 오른쪽 그림은 관측치의 값과는 상관 없이 회귀 선이 데이터에 완벽하게 적합된다. 이것은 데이터에 과적합될 우려의 문제가 있다. 다시 말해서 훈련 데이터에 대해서 고차원 세팅이 잘 작동하겠지만 독립적인 테스트 세트에 대해서는 매우 좋지 않은 결과를 낼 것이고 그에 따라서 좋은 모델을 형성할 수 없다. 즉, $p > n$ 또는 $p \approx n$ 이라면 너무 flexible해서 데이터에 과적합될 우려가 있다.

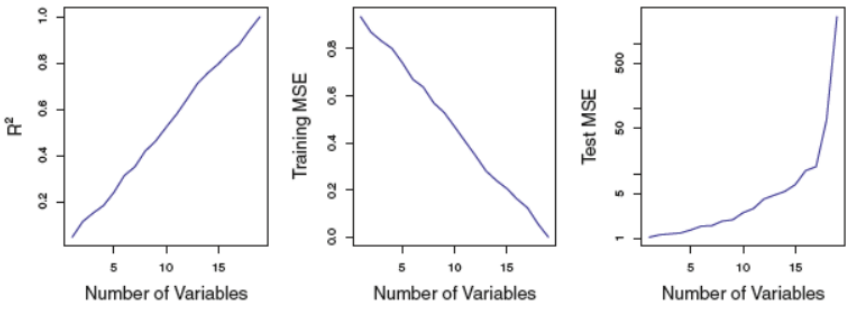


FIGURE 6.23. *On a simulated example with $n = 20$ training observations, features that are completely unrelated to the outcome are added to the model.* Left: *The R^2 increases to 1 as more features are included.* Center: *The training set MSE decreases to 0 as more features are included.* Right: *The test set MSE increases as more features are included.*

figure 6.23은 고차원 데이터를 주의 깊게 다루지 않을 시에 발생하는 위험에 대해서 설명한다. 데이터의 관측치와 변수는 각각 20개인데 설명변수가 반응변수와 전혀 관련이 없게

구성되어 있다. 설명 변수가 반응변수와 전혀 상관이 없음에도 변수의 갯수가 늘어날 수록 결정계수는 1에 가까워지고 training MSE는 0에 가까워진다. 하지만 독립적인 test set에 대한 test MSE는 변수의 갯수가 늘어날수록 매우 높아지는데 그 이유는 계수 추정치의 부산이 추가적인 설명변수를 포함할수록 매우 커지기 때문이다. test MSE를 보면 가장 좋은 모델은 변수를 적게 포함하는 모델임이 분명하다. 이를 통해서 고차원 데이터를 다룰 때 항상 주의해야 하고 모델을 독립적인 test set에 대한 성능을 점검해보는 것이 매우 중요함을 알 수 있다.

우리가 6.1.3에서 살펴보았던 C_p, AIC, BIC는 고차원 데이터에 대한 적절한 방법이 될 수 없는데 그 이유는 $\hat{\sigma}^2$이 문제를 일으키기 때문이다.(예를 들어 Chapter 3의 공식을 사용하면 얘의 추정치는 0이 이다.) 마찬가지로 adjusted R^2도 1의 값을 쉽게 얻을 수 있기 때문에 문제가 될 수 있다. 대안적인 방법이 필요해 보인다!

6.4.3 Regression in High Dimensions

우리가 해당 챕터에서 배웠던 덜 flxible한 모델을 적합하는 방법인 forward stepwise selection, ridge, lasso, principal components regression은 고차원 데이터에서 회귀를 시행하는데 특히 유용하다. 이러한 접근법은 least squares보다 덜 flexible한 접근법으로 과적합을 막는데 효과적이다.

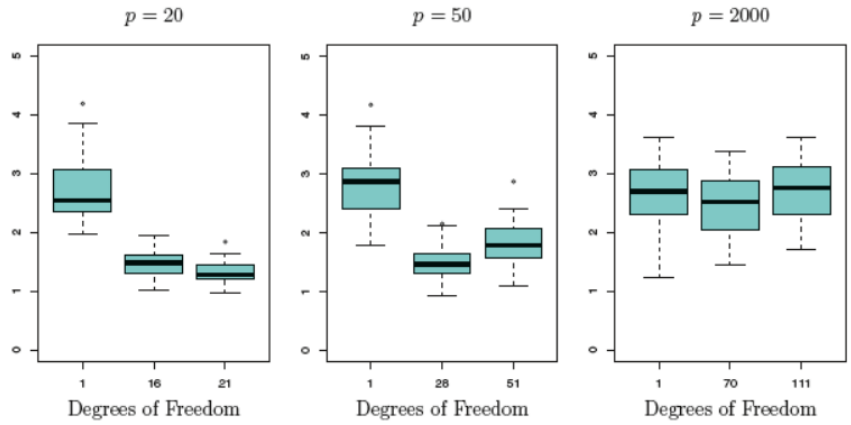


FIGURE 6.24. *The lasso was performed with $n = 100$ observations and three values of p, the number of features. Of the p features, 20 were associated with the response. The boxplots show the test MSEs that result using three different values of the tuning parameter λ in (6.7). For ease of interpretation, rather than reporting λ, the* degrees of freedom *are reported; for the lasso this turns out to be simply the number of estimated non-zero coefficients. When $p = 20$, the lowest test MSE was obtained with the smallest amount of regularization. When $p = 50$, the lowest test MSE was achieved when there is a substantial amount of regularization. When $p = 2{,}000$ the lasso performed poorly regardless of the amount of regularization, due to the fact that only 20 of the 2,000 features truly are associated with the outcome.*

figure 6.24는 lasso 결과를 보여준다. 여기서 변수는 20개, 50개, 2000개이고 20개만 반응변수와 관련이 있다. training data 100개에 대해서 진행되었고 test MSE가 측정이 되었다. $p = 20$일 때, λ가 작을 때 test error가 가장 작았다.(위 그림에서는 자유도(degrees of freedom)으로 표시됨. lasso에서 이것은 간단하게 추정된 계수가 0이 아닌 애들의 수이고 lasso 적합의 flexibility를 측정하는 수단이다. 따라서 자유도가 높다는 것은 계수가 0이 아닌 애들이 많다는 것이고 이것은 λ가 작다는 뜻일듯.) $p = 50$일 때는 regrlarization이 상당하게 있을 때 test MSE가 가장 낮았고 $p = 2000$일 때는 regularization과 상관 없이 결과가 좋지 않았다.

figure 6.24는 세가지 중요한 점을 알려준다. regularization 또는 shrinkage는 고차원 문제에 중요한 역할을 하고 적절한 tuning parameter selection은 예측력있는 모델을 위해 중요하며 추가되는 변수가 반응변수와 연관되어 있지 않는 이상, test error는 증가하는 경향이 있다. 위의 세 번째 포인트는 고차원 데이터에서 핵심 원칙이며 이것을 curse of dimensionality 라고 부른다. 언뜻 보면 변수가 추가될 수록 더 좋은 모델이 만들어질 것 같지만 사실은 그 반대이다. 물론 추가되는 변수가 반응 변수가 진정으로 연관되어 있다면 test set error을 줄여서 적합된 모델의 성능을 향상시키겠지만 noise 변수는 모델의 성능을 악화시키고 결과적으로 test set error을 증가시킨다. 왜냐하면 noise 변수는 데이터의 차원을 증가시키고 과정합의 위험을 증가시키기 때문이다.

6.4.4 Interpreting Results in High Dimensions

고차원 데이터에서 multicollinearity(다중공선성) 문제는 매우 심하다. 특히 고차원 데이터에서 다중공선성의 문제가 더 심각할 수 있어서 어떤 변수도 다른 모든 변수의 linear combinations으로 다시 표현될 가능성이 있다. 따라서 하나의 좋아 보이는 모델을 만들었다고 하더라도 그 모델은 그저 가능한 많은 모델 중 하나이고 독립적인 데이터 셋에 대해서 더 검증되어야 함을 명심해야 한다.

또한 고차원 데이터에서 error나 모델의 적합도에 대해서 얘기할 때 특히 주의해야 한다. $p > n$일 때, 잔차가 0인 쓸모 없는 모델을 얻기 쉬움을 앞서 살펴보았다. 따라서 절대로 sum of squared errors, p-values, R^2 또는 다른 전통적인 모델의 적합을 측정하는 방법을 고차원 데이터에서 모델의 적합도를 보여주는 방법으로 사용해서는 안된다. 예를 들어, figure 6.23에서 $R^2 = 1$인 모델을 쉽게 보았다. 이것은 쓸모없는 모델인데도 다른 사람들이 좋은 모델이라고 오해하게 할 수 있다. 고차원 데이터에 대해서는 대신에, 독립적인 test set에 대한 결과 또는 cross-validation errors에 대해서 말하는 것이 좋다. 예를 들어 독립적인 test set에 대한 MSE나 R^2이 training set에 대한 MSE보다 모델 적합도를 측정할 때 더욱 효과적인 방법이다.