Dimension Reduction Methods

p개의 변수를 가지는 모형을 M($<$ p)개의 변수를 가지는 모형으로 차원을 축소시키는 방법.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$$

$$Transformation\ Z_m = \sum_{j=1}^{p} \phi_{jm} X_j\ where\ \phi_{jm}\ are\ constant\ and\ M < p$$

$$\begin{aligned} Y &= \theta_0 + \theta_1 Z_1 + \theta_2 Z_2 + \cdots + \theta_M Z_M + \epsilon \\ &= \theta_0 + \sum_{m=1}^{M} \theta_m z_m + \epsilon \end{aligned}$$

여기서 차원 축소 접근을 원래 모형의 계수들에 대해 제약(constraints)을 두는 것이라고 볼 수 있다.

$$\sum_{m=1}^{M} \theta_m z_m = \sum_{m=1}^{M} \theta_m \sum_{j=1}^{p} \phi_{jm} x_j = \sum_{j=1}^{p} \sum_{m=1}^{M} \theta_m \phi_{jm} x_j = \sum_{j=1}^{p} \beta_j x_j$$

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

이는 bias-variacne trade off 관점에서 잠재적인 bias의 가능성이 있지만 $M << p$인 M을 고른다면 분산을 상당히 많이 줄여줄 것이다.

An Overview of Principal Components

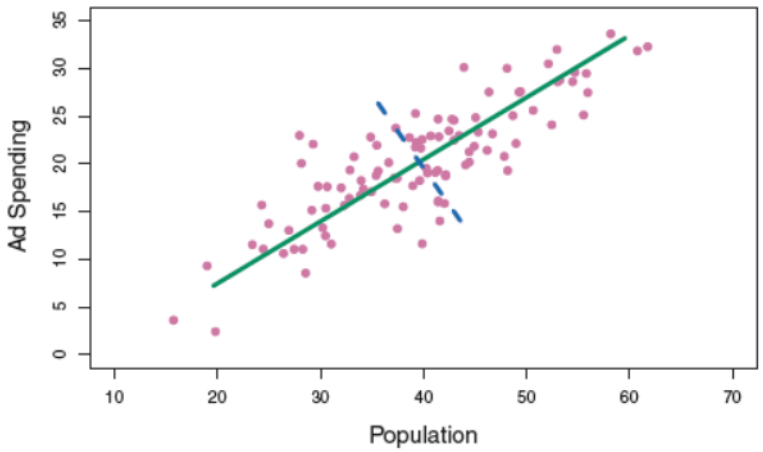
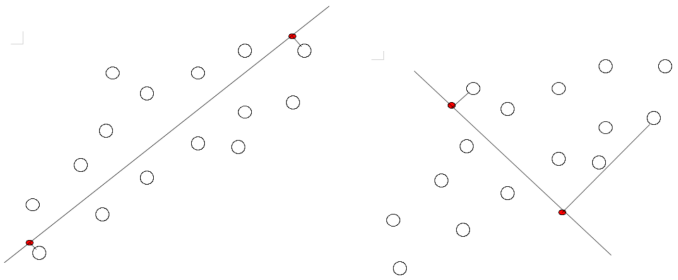


FIGURE 6.14. *The population size (pop) and ad spending (ad) for 100 different cities are shown as purple circles. The green solid line indicates the first principal component, and the blue dashed line indicates the second principal component.*

first principal component: 주성분으로 데이터를 사영 (projection) 했을 때, 가장 큰 분산을 가지는 백터.



제약조건 : $\phi_{11}^2+\phi_{21}^2 = 1$을 만족하는 pop과 ad의 모든 선형 조합에서 위 principal component는 가장 큰 분산을 가진다. $\phi_{11}^2 + \phi_{21}^2 = 1$ 조건이 없다면 분산을 최대화하기 위해 ϕ_{11}, ϕ_{21}을 임의적으로 늘릴 수 있기 때문이다.

초록색 선 (first principal component): $Z_1 = 0.839 \times (pop - \overline{pop}) + 0.544 \times (ad - \overline{ad})$

principal component loadings: $\phi_{11} = 0.839, \phi_{21} = 0.544$ 위의 directions을 정의한다.

Z_1, pop, ad 는 모두 길이가 100인 (관측치의 갯수가 100개 이므로) 백터다. 따라서 개별 값은 다음과 같다.

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

$z_{11}, z_{21}, \cdots, z_{n1}$은 principal component scores이다.

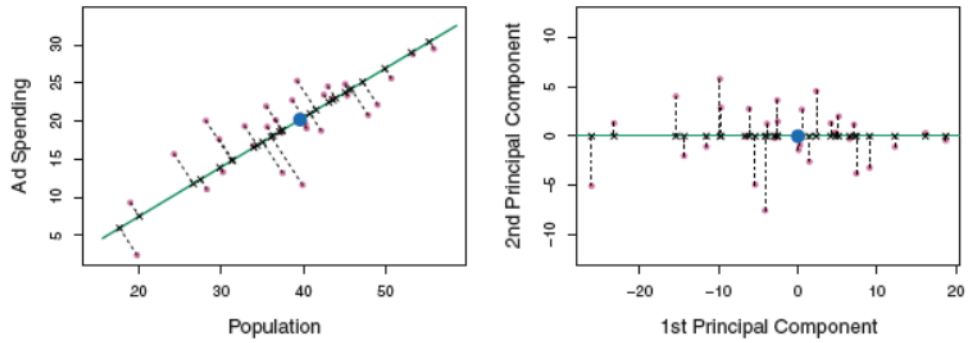


FIGURE 6.15. *A subset of the advertising data. The mean* pop *and* ad *budgets are indicated with a blue circle.* Left: *The first principal component direction is shown in green. It is the dimension along which the data vary the most, and it also defines the line that is closest to all* n *of the observations. 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.*

principal component scores: 관측치와 first principal component 선 사이의 거리로도 볼 수 있다.

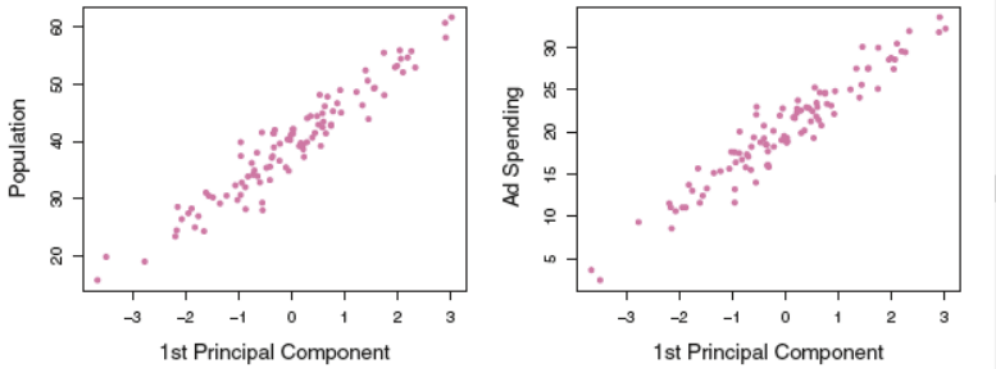


FIGURE 6.16. *Plots of the first principal component scores z_{i1} versus* pop *and* ad*. The relationships are strong.*

위의 그림은 first principal component와 pop, Ad간의 관계를 그래프에 표현한 것이다. 위의 그림을 통해 두 변수는 first principal component와 강한 선형의 관계가 있음을 확인할 수 있다. 즉, 다시 말해 두 변수를 잘 설명하는 principal component을 뽑은 것이다.

Second principal component: 마찬가지로 변수들의 여러가지 선형결합에서 first principal component와 관계가 없어야(uncorrelated) 하고 가장 큰 분산을 가진다.

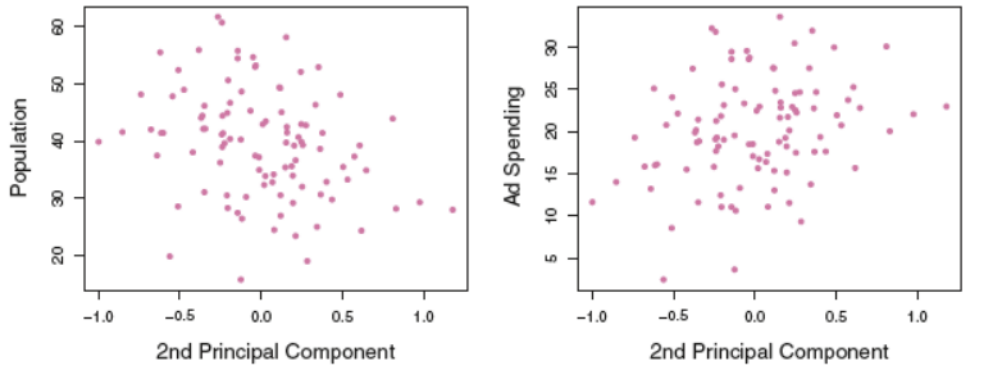


FIGURE 6.17. *Plots of the second principal component scores z_{i2} versus* pop *and* ad*. The relationships are weak.*

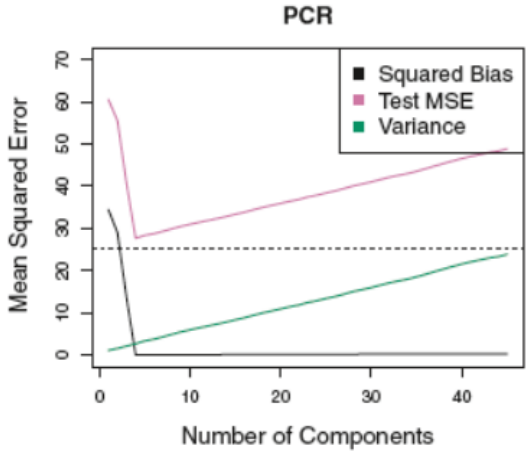
second principal component와 pop, ad와의 관계가 거의 나타나지 않는다. 이를 통해 first principal component가 pop, ad의 대부분의 정보를 설명함을 알 수 있다.

자세한 내용은 10장에서.....

The principal Components Regression Approach

PCR은 위에서 변환된 $Z_1, \cdots, Z_M$ 을 예측변수로써 사용하여 선형회귀를 실시하는 접근법이다. 선형회귀가 예측변수와 반응변수간에 선형의 관계가 있다고 가정하는 것과 유사하게 PCR도 principal component을 나타내는 direction이 반응변수와 연관이 있다고 가정한다.

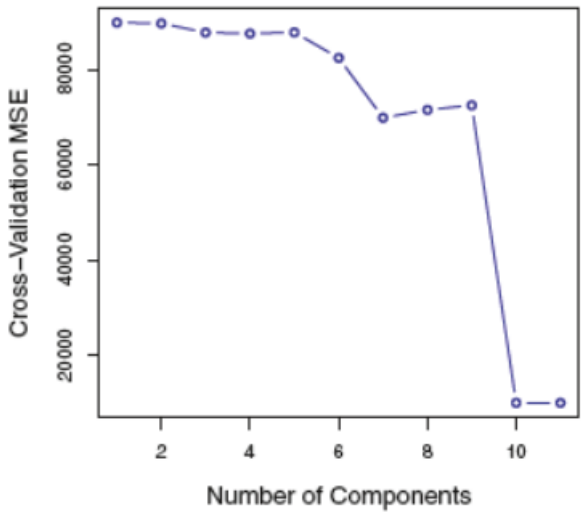
언제 PCR을 쓰면 좋을까?



PCR이 좋은 성능을 보일 때: 처음의 적은 principal components로 반응변수의 많은 부분을 설명할 수 있는 데이터

PCR이 나쁜 성능을 보일 때: 적절한 모델을 만들기 위해 많은 principal components가 필요한 데이터

M의 선택... cross-validation을 통해서.



PCR을 시행할 때, 표준화(standardizing)을 하는 것이 좋다.(scaling을 위해) 하지만 표준화 없이 단위를 바꿀 수 있는 경우, 굳이 표준화를 할 필요는 없다.

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

Partial Least Squares

PCR은 예측 변수의 선형 결합인 principal component가 반응변수를 가장 잘 설명할 것이라고 가정한다. 이것은 비지도 학습(unsupervised learning)의 예로써, 사실 반드시 잘 설명하리라는 보장은 없다. 이러한 PCR의 결점을 보완하는 것이 바로 지도 학습(supervised learning)인 PLS이다.

PLS의 Z_1(first direction)을 계산하는 과정은 다음과 같다.

1. p개의 변수를 표준화한 후, PLS는 각각의 ϕ_{j1}을 X_j로의 Y 선형 회귀에서의 계수와 동일하게 설정한다. 이 계수는 Y, X_j의 상관계수에 비례한다.
2. 그리고 $Z_1 = \sum_{j=1}^{p} \phi_{j1} X_j$를 계산할 때, PLS는 반응변수와 가장 강하게 연관되어 있는 변수에 가장 높은 가중치를 둔다. (∵1번)

PLS의 Z_2(second direction)을 계산하는 과정은 다음과 같다.

1. $Z_1 = \sum_{j=1}^{p} \phi_{j1} X_j$에서 각각의 변수들을 Z_1에 대해 회귀를 실시한 후, 잔차(residuals)을 구한다.

2. 이 잔차는 Z_1에 의해서 설명되지 않은 정보들로 해석할 수 있고 이것을 사용하여, Z_1을 구했던 절차와 동일하게 Z_2을 계산한다. 이러한 과정을 M번 반복하여 PLS components인 $Z_1, \cdots, Z_M$을 구한다.

마지막으로 $Z_1, \cdots, Z_M$을 예측변수로 least squares을 이용해 Y을 예측하는 선형 모델을 적합시킨다.

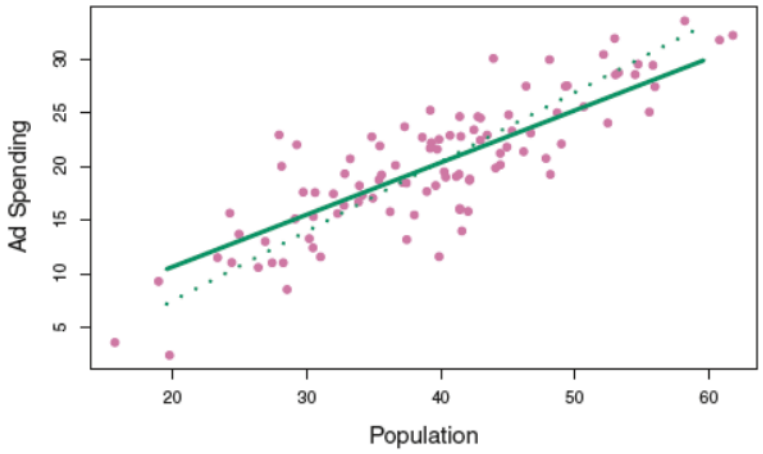


FIGURE 6.21. *For the advertising data, the first PLS direction (solid line) and first PCR direction (dotted line) are shown.*

점선은 PCR direction이고 실선은 PLS direction이다. 여기서 실선이 점선보다 기울기가 작은데, 이는 Ad에 대해서 변화가 적음을 알 수 있다.(pop이 한 단위 변할 때, Ad의 변화량은 PCR보다 PLS가 더 작다) 이는 PLS에서는 Ad보다 pop이 반응변수와 더 연관되어 있다고 판단했기 때문이다.

High Dimensional Data

관측치의 갯수인 n이 변수의 갯수인 p에 비해서 작을 때, 고차원이라고 말한다. 고차원 데이터에 대해서 least squares regression이나 로지스틱 회귀 같은 고전적인 기법들은 효과적이지 못하다. 아래를 살펴보자.

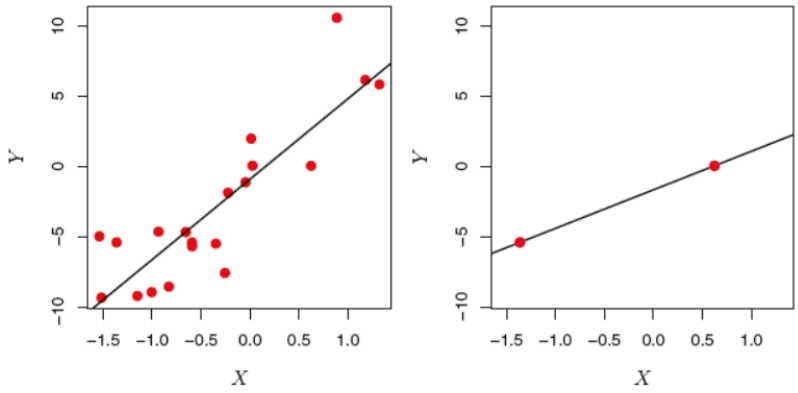


FIGURE 6.22. Left: *Least squares regression in the low-dimensional setting.* Right: *Least squares regression with $n = 2$ observations and two parameters to be estimated (an intercept and a coefficient).*

위 그림에서 관측치가 매우 적은 오른쪽 그림은 잔차가 0인, training data에 대해서 완벽하게 적합이 되었다. 하지만 이는 과적합(overfitting)의 위험이 있으며 training data와 독립적인 test set에 대해서는 해당 모델이 너무 유연하기(flexible) 때문에 좋은 결과를 내지 못할 것이다.

그렇다면 어떤 모델을?!

고차원 데이터에서 특히 모델의 유연성(flexibility)이 높아지는 것에 유의해야 하므로 앞서 배운 forward stepwise selection, ridge, lasso, principal components regression이 효과적이다.

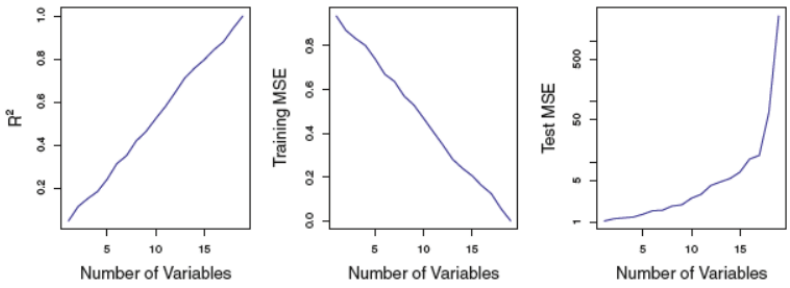


FIGURE 6.23. *On a simulated example with $n = 20$ training observations, features that are completely unrelated to the outcome are added to the model.* Left: *The R^2 increases to 1 as more features are included.* Center: *The training set MSE decreases to 0 as more features are included.* Right: *The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

위 그림에서 사용된 데이터의 변수 20개는 반응변수와 전혀 관련이 없는 변수들이다. 하지만 변수의 갯수가 증가할수록 결정계수는 1에 가까워지고 training MSE는 0에 가까워 지고 있지만 test MSE는 급등하고 있다.

curse of dimension: 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 악화시키기 때문이다.

고차원 데이터에서 multicollinearity(다중공선성) 문제는 매우 심하다. 어떤 변수도 다른 모든 변수의 linear combinations으로 다시 쓸 수 있다. 이것은 어떠한 변수가 반응 변수와 관련이 되어 있는지 (만약 관련이 있다면) 절대로 알 수 없고 회귀에서 가장 좋은계수를 알아낼 수 없다. 따라서 하나의 좋아 보이는 모델을 만들었다고 하더라도 그 모델은 그저 가능한 많은 모델 중 하나이고 독립적인 데이터 셋에 대해서 더 검증되어야 함을 명심해야 한다

그렇다면 더 나은 지표는?!

training data에 대해서만 모델의 정확도를 측정하는 것이 아니라 반드시 독립적인 test set에 대한 MSE, 또는 cross-validation error를 이용하여 모델의 정확도를 말하는 것이 좋다.