Lecture16. Projection Matrices and Least Squares

▼ 목차

1. 투영(Projection)

1) Two extreme cases

Lecture 15에 나왔던 내용 중 **투영 행렬**(Projection matrix)에 대해 다시 살펴보자.

투영 행렬은 어떤 벡터를 다른 공간으로 투영시키는 행렬이고 식은 다음과 같다.

$$P = A(A^TA)^{-1}A^T$$

투영 행렬은 곱해지는 벡터 b를 column space에 존재하는 벡터 중 가장 가까운 점이나 가장 가까운 벡터로 바꿔주는 역할을 한다.

만약 벡터 b가 이미 column space에 존재한다면? 혹은 벡터 b가 column space와 수직이라면? 어떤 결과가 발생할까?

벡터 b가 이미 column space에 존재하는 경우를 살펴보자. 이때는 투영 행렬 P에 b를 곱하면 그대로 b를 얻는다.

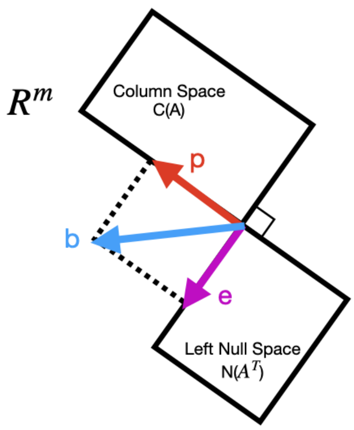
벡터 b는 A의 column space에 존재하므로 A의 열들의 선형결합이다. 즉, b = Ax로 나타낼 수 있고, Ax를 column space에 투영하면 다음과 같다.

$$p = Pb = A(A^TA)^{-1}A^TAx = Ax = b$$
$$\therefore Pb = b$$

다음으로 벡터 b가 column space와 수직인 경우를 살펴보자. 이 경우에는 투영 행렬 P와 b를 곱하면 0이 된다.

이전에 배웠던 부분공간(subspace)을 생각해보면 이유를 알 수 있다. Pb=0이 된다는 것은 b가 column space가 아닌 Left null space에 존재한다는 것이다. Lecture 14에서 Left null space는 column space와 수직임을 배웠다. 따라서 column space에 수직인 b는 곧 Left null space에 존재하는 것이다.

아래의 사진에서 파란색 벡터 b를 column space에 투영한 벡터가 바로 빨간색 벡터 p가 되고, 반대로 Left null space에 투영시키면 보라색 벡터e가 된다.



투영행렬 P를 b에 곱하여 p가 되고 단위 행렬에서 투영행렬을 뺀 (I-P)행렬에 b를 곱해서 e가 된다.

이때 **(I-P)는 P행렬로 투영시키는 공간과 직교하는 공간으로 투영시키는 역할을 한다.**

(I-P)가 투영행렬이 되는 이유: 투영행렬 P는 대칭(symmetric)이며 $P^2 = P$이다. 다음과 같이 $(I-P)$도 대칭이며, $(I-P)^2 = (I-P)$를 만족하기 때문에 투영행렬이라고 할 수 있다.

$$(I-P)^T = I^T - P^T = I - P^T = I - P$$
$$(I-P)^2 = I^2 - 2P + P^2 = I - 2P + P = I - P$$

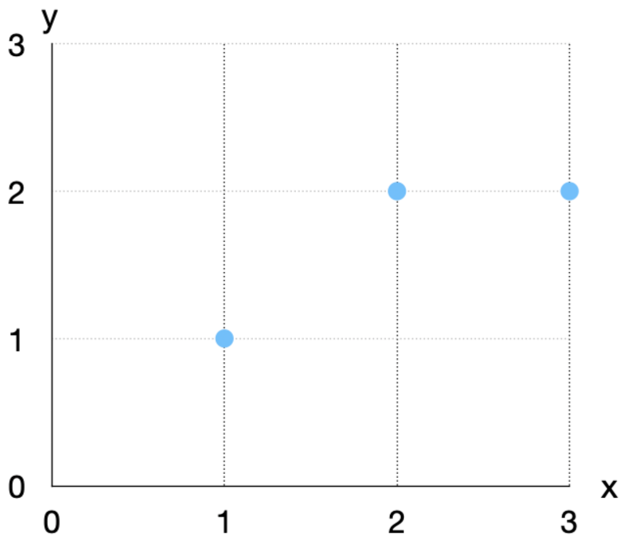
2. 최소제곱법(Least Square Method)

최소제곱법 또는 최소자승법이라고 하는 Least Square Method에 대해서 알아볼 것이다. 회귀분석, 수치해석 등 다양한 분야에서 쓰이기 때문에 잘 알아두면 좋다!

1) Basics of Least Square

최소제곱법(Least Square Method)은 수집한 데이터를 기반으로 데이터들을 최대한 만족시키는 하나의 직선에 대한 식을 찾는 방법이다. 이렇게 찾은 직선을 통해 데이터에 대한 예측 뿐만 아니라 시스템의 특성까지 파악할 수 있다.

아래와 같이 (1,1) (2,2) (3,2) 3개의 데이터를 얻었다고 해보자.



이 데이터의 경향을 잘 나타내는 직선을 찾아야한다. 우리가 잘 아는 일차방정식 y=gx+h에서 g와 h를 찾으면 직선을 구할 수 있다.

y=gx+h를 Ax=b의 꼴로 나타내고 투영 행렬의 해를 구했을 때처럼 x에 대해서 풀어주면 된다. 즉, 주어진 데이터를 Ax=b의 식으로 표현하면 된다. 우선 주어진 데이터를 가지고 다음과 같은 3개의 방정식을 만들어보자.

$$\begin{gathered} gx + h = y \\ \Downarrow \\ g + h = 1 \\ 2g + h = 2 \\ 3g + h = 2 \end{gathered}$$

위의 3개의 식을 Ax=b꼴로 만들면 다음과 같다.

$$\underset{A}{\begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{bmatrix}} \underset{\mathrm{x}}{\begin{bmatrix} g \\ h \end{bmatrix}} \underset{=}{=} \underset{b}{\begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}}$$

위의 식은 미지수보다 방정식이 많은 overdetermined case이다.(Lecture15참고) 행렬 A의 column1과 column2는 독립(independent)이며 또한 column space의 기저(basis)이다. b가 column space에 존재해야 해가 존재하는데 b는 column space에 존재하지 않는다. 따라서 이 식을 완벽하게 만족시키는 해 x는 존재하지 않는다. 다만 가장 근사한 해, best solution을 구할 뿐이다. Lecture15에서 배웠듯이 근사한 해를 구하는 방법은 Ax=b의 양변에 A transpose를 각각 곱해주면 된다.

$$A^T A\hat{x} = A^T b$$

모든 데이터를 지나는 직선이 없기 때문에 **각 데이터와 직선 사이의 에러의 총합이 최소가 되는 직선을 찾아야 한다.**

그렇다면 **에러**(error)는 무엇일까? 방정식에서 g와 h의 값을 추정했다고 했을 때, 3개의 방정식에 대입하면 각각의 $\hat{y}$(y의 추정치)가 나올 것이다. 이 $\hat{y}$값과 실제 y값 사이의 차이가 에러(error)이다. 이를 다음과 같이 표현할 수 있다.

$$\text{error } e = (\hat{y} - y)^2 = (\hat{g}x + \hat{h} - y)^2$$

에러 값이 음수가 나오는 경우가 있는데 이 경우 값이 상쇄되어 에러 값이 0이 나오는 경우가 있다. 이렇게 음수 값이 나오지 않도록 항상 에러 값을 제곱해준다. 제곱이 아니라 4제곱, 6제곱, $\cdots$, 2n제곱을 해도 되지만 제곱이 제일 계산하기 편해서 제곱을 사용한다. 절댓값도 가능하지만 마찬가지로 계산이 어렵다.

모든 방정식의 에러 값을 더한걸 최소로 하는 g와 h를 찾는것이 목표이다!

$$\text{Minimize } \sum_{j=1}^{m} e^2 = \sum_{j=1}^{m} (gx_j + h - y_j)^2$$

위의 예시의 total error는 다음과 같다.

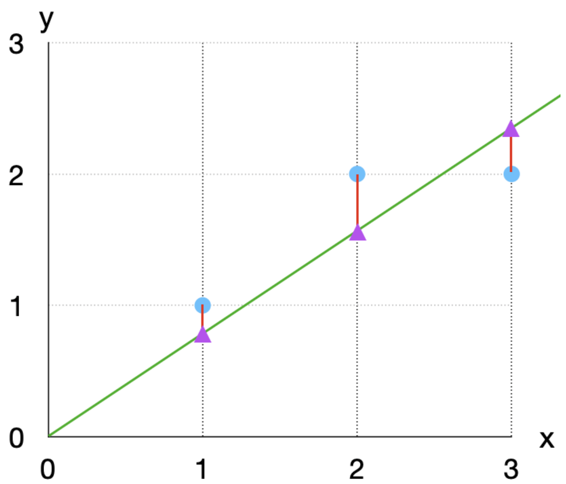
$$total\ error = (g + h - 1)^2 + (2g + h - 2)^2 + (3g + h - 2)^2$$

위에서 방정식을 Ax=b의 꼴로 만들었다. 에러를 최소화 시키는 식을 행렬로 정리하면 식은 아래와 같이 될 것이다.

$$\text{Minimize } ||Ax - b||^2 = ||e||^2$$

> 📌 **최소제곱법**
> 오차의 제곱의 합을 최소화해서 파라미터를 찾는 방법

이렇게 구한 직선을 그림으로 그려보자.



초록색 직선이 우리가 위에서 구한 직선이고 파란색 점은 원래 값, 보라색 세모는 추정치, 빨간 선은 에러를 의미한다.

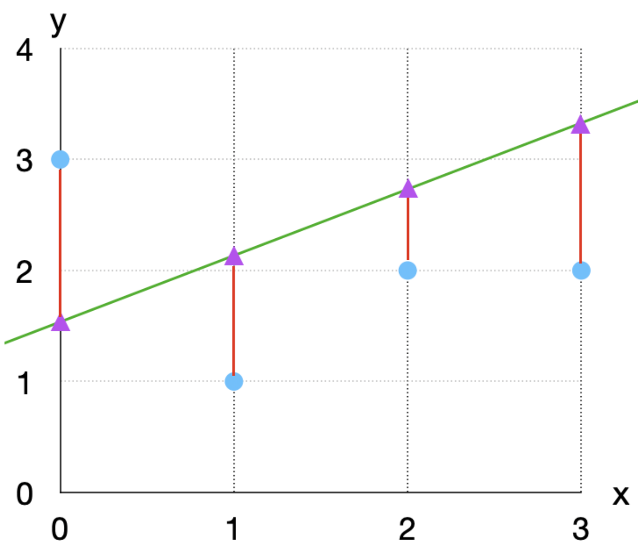
주어진 데이터를 이용하여 데이터들의 추세를 파악하고 예측한다. 이것을 **회귀(regression)**라고 한다.

결국 최소제곱법은 주어진 데이터를 이용해 회귀를 한 것이다. 여기서는 직선을 fitting했기 때문에 선형회귀를 한 것이다.

2) Outliers in Least square

실제 데이터를 가지고 추세를 파악할때 쓸모없는 값인 노이즈 데이터들이 많다. 대부분의 데이터들의 분포에서 큰 거리에 떨어져있는 소수의 데이터들을 **이상치(Outlier)**라고 하는데 대부분 노이즈일 경우가 많다.

만약 위의 데이터에서 이상치가 추가된다면 직선은 아래의 그림처럼 아까와는 매우 다를 것이다.



위의 그림은 (0,3)이라는 이상치가 존재할 경우의 직선이다. 이상치의 에러도 고려하다보니 아까 구했던 직선과 매우 다른 모습의 직선이 나온다. 결국 이상치 때문에 전체적인 데이터의 경향을 고려하지 못하게 되었다. 이를 overcompensate이라 한다.

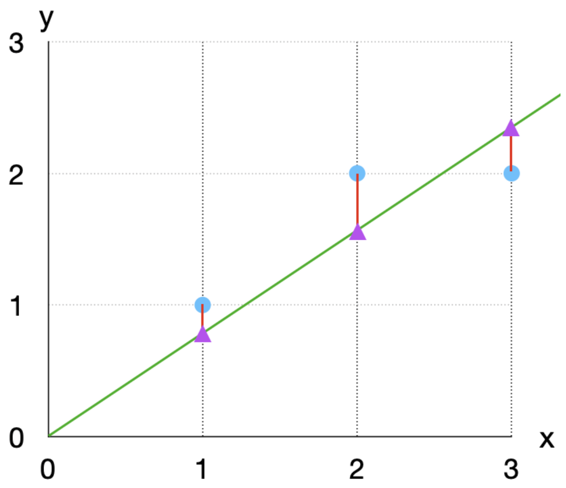
3) Solution of Least square

위에서 살펴봤던 식을 다시 살펴보자.

$$\underset{A}{\begin{bmatrix}1 & 1\\2 & 1\\3 & 1\end{bmatrix}}\underset{\mathrm{x}}{\begin{bmatrix}g\\h\end{bmatrix}} \underset{=}{=} \underset{b}{\begin{bmatrix}1\\2\\2\end{bmatrix}}$$

행렬 A의 column vector는 독립(independent)이며 column space의 기저(basis)이다. b벡터는 A의 선형 결합으로 표현할 수 없기 때문에 이 column space에 존재하지 않는다. 따라서 Ax=b의 해는 존재하지 않는다. 정확한 해는 존재하지 않지만 근사한 해를 구할 수 있다. 이를 구하기 위해서 b를 column space에 투영시키면 된다.

b를 column space로 투영시킨 벡터를 p라고 하면 아래의 그림에 존재하는 보라색 세모들은 벡터 p=[p1 p2 p3]가 된다.



벡터 p는 A의 column space로 투영시킨 것이기 때문에 **해가 존재**한다. b를 p로 투영시켰기 때문에 식은 아래와 같이 바뀐다.

$$A\hat{\mathrm{x}} = p$$
$$\Downarrow$$
$$\begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} \hat{g} \\ \hat{h} \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

정확한 해가 아니라 최적해(best solution)이기 때문에 x가 아니라 $\hat{\mathrm{x}}$이다.

이제 $\hat{\mathrm{x}}$를 구해보자. Ax=b 양변에 A transpose를 곱해주고 그것을 다시 방정식으로 표현해보면 다음과 같다.

$$A^T A\hat{\mathrm{x}} = A^T b$$
$$\Downarrow$$
$$\begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} \hat{g} \\ \hat{h} \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$$
$$\begin{bmatrix} 14 & 6 \\ 6 & 3 \end{bmatrix} \begin{bmatrix} \hat{g} \\ \hat{h} \end{bmatrix} = \begin{bmatrix} 11 \\ 5 \end{bmatrix}$$
$$\Downarrow$$
$$14\hat{g} + 6\hat{h} = 11$$
$$6\hat{g} + 3\hat{h} = 5$$

이전에 배웠던 가우스 조던 소거법(Gauss-Jordan elimination)을 이용하여 $\hat{g}$, $\hat{h}$를 구해보자.

$$\left[\begin{array}{cc|c} 14 & 6 & 11 \\ 6 & 3 & 5 \end{array}\right] \xrightarrow[\frac{1}{14}\times row1]{} \left[\begin{array}{cc|c} 1 & \frac{3}{7} & \frac{11}{14} \\ 6 & 3 & 5 \end{array}\right] \xrightarrow[(-6)\times row1 + row2]{} \left[\begin{array}{cc|c} 1 & \frac{3}{7} & \frac{11}{14} \\ 0 & \frac{3}{7} & \frac{2}{7} \end{array}\right] \xrightarrow[\frac{7}{3}\times row1 + row2]{} \left[\begin{array}{cc|c} 1 & \frac{3}{7} & \frac{11}{14} \\ 0 & 1 & \frac{2}{3} \end{array}\right]$$

위와 같이 소거법을 진행한 후에 후방 대입법(back substitution)을 통해 $\hat{h} = \frac{2}{3}$, $\hat{g} = \frac{1}{2}$임을 알 수 있다.

따라서 최적의 직선의 방정식은 다음과 같다.

KaTeX parse error: Unexpected character: ' ' at position 1: p=\frac{1}{2}x+…

여기서 에러는 어떻게 구할 수 있을까? 에러는 e=y-p이다.

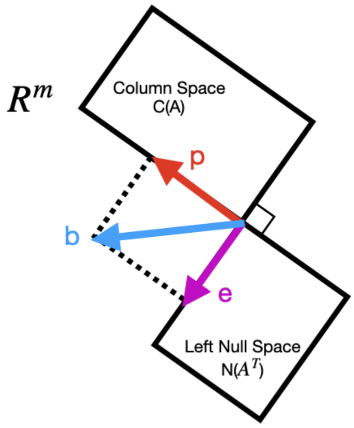
위에서 구한 직선에 방정식에 x=1,2,3을 넣은 값 p와 실제 y값을 각각 빼주면 된다.

$$p = \begin{bmatrix} \frac{7}{6} \\ \frac{5}{3} \\ \frac{13}{6} \end{bmatrix}, e = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} - \begin{bmatrix} \frac{7}{6} \\ \frac{5}{3} \\ \frac{13}{6} \end{bmatrix} = \begin{bmatrix} -\frac{1}{6} \\ \frac{2}{6} \\ -\frac{1}{6} \end{bmatrix}$$

이러한 방법으로 주어진 데이터들을 이용해 방정식을 세우고 최소제곱법을 이용해 이 데이터들을 가장 근사하는 직선을 찾을 수 있다.

4) Error vector

에러 벡터에 대해서 더 알아보자. 에러 벡터에 투영벡터 p를 더하면 원래 벡터인 y가 된다. 초반에 나왔던 아래의 그림처럼 p와 e는 직교(perpendicular)이다.



실제로 그러한지 위의 예시를 가지고 확인해보자.

우선 에러 벡터에 투영벡터 p를 더하면 원래 벡터인 y가 되는 것을 다음과 같이 쉽게 확인 할 수 있다.

$$y = e + p$$
$$\begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{6} \\ \frac{2}{6} \\ -\frac{1}{6} \end{bmatrix} + \begin{bmatrix} \frac{7}{6} \\ \frac{10}{6} \\ \frac{13}{6} \end{bmatrix}$$

p와 e가 직교(perpendicular)하는지 알아보기 위해 둘을 내적해보면 $\frac{-7}{36} + \frac{20}{36} - \frac{13}{6} = 0$이 된다. 내적을 통해 p와 e가 직교하는 것을 확인할 수 있다.

그림을 다시 보면 e는 p뿐만 아니라 A의 column space와도 직교한다. 실제로 다음과 같이 내적을 계산해보면 0이 되어 직교하는 것을 확인할 수 있다.

$$e^T = col1$$
$$\begin{bmatrix} -\frac{1}{6} & \frac{2}{6} & -\frac{1}{6} \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$
$$e^T = col2$$
$$\begin{bmatrix} -\frac{1}{6} & \frac{2}{6} & -\frac{1}{6} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

최소제곱법은 아래의 핵심 방정식을 이해하는 것이 중요하다.

$$A^T A\hat{x} = A^T b$$
$$A\hat{x} = p$$

이때 $A^T A\hat{x} = A^T b$를 $\hat{x}$으로 정리하기 위해서 $A^T A$가 역행렬이 존재해야한다. 역행렬이 존재하려면 A의 column vector들이 독립(independent)해야 한다. 독립이 아니라면 역행렬이 존재하지 않고 결국 해를 구할 수 없다.