Notes on Linear Systems and Matrix Decompositions

Gyubeom Edward Im*

February 8, 2024

Contents

*blog: alida.tistory.com, email: gyurse@gmail.com

1 Linear equation

선형방정식 (Linear Equation)은 변수 $x_1, \cdots, x_n$이 있을 때 다음과 같이 작성할 수 있는 방정식을 의미한다.

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b \tag{1}$$

이 때, b는 계수를 의미하고 $a_1, \cdots, a_n$ 값들은 실수 또는 복소수의 미지수를 의미한다. 위 식은 다음과 같이 간결하게 작성할 수 있다.

$$\mathbf{a}^\intercal\mathbf{x} = b \tag{2}$$

이 때, $\mathbf{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}$ 이고 $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ 이다.

2 Linear system

선형방정식(linear equation)의 집합을 선형시스템(linear system)이라고 한다. n개의 선형방정식 $\mathbf{a}_1\mathbf{x} = b_1, \cdots, \mathbf{a}_n\mathbf{x} = b_n$ 이 있는 경우 이를 다음과 같이 간결하게 선형시스템으로 표현할 수 있다.

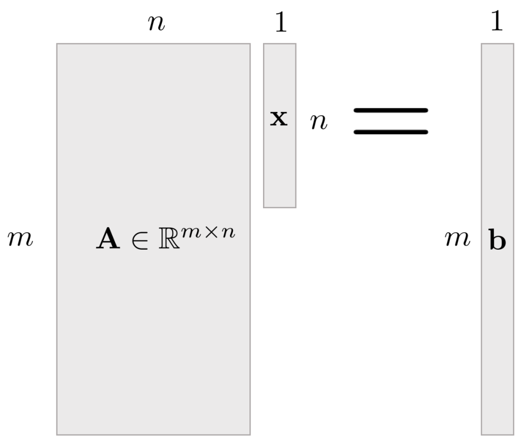
$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{3}$$

즉, $\mathbf{A}\mathbf{x} = \mathbf{b}$ 형태의 행렬과 벡터의 방정식을 선형시스템이라고 한다. 선형시스템은 다른 말로 비동차방정식이라고도 불린다. 동차방정식, 비동차방정식의 정의는 다음과 같다.

2.1 Homogeneous equation

동차방정식(homogeneous equation)은 $\mathbf{A} \in \mathbb{R}^{n\times m}, \mathbf{x} \in \mathbb{R}^{m\times 1}, \mathbf{b} \in \mathbb{R}^{m\times 1}$ 일 때, $\mathbf{A}\mathbf{x} = 0$ 형태의 시스템을 말한다. 0이 아닌 해가 존재한다. 이와 반대로 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 형태의 방정식을 비동차방정식(non-homogeneous equation)이라고 한다. 비동차방정식은 해가 존재하지 않거나 여러 개 존재한다.

2.2 Over-determined system

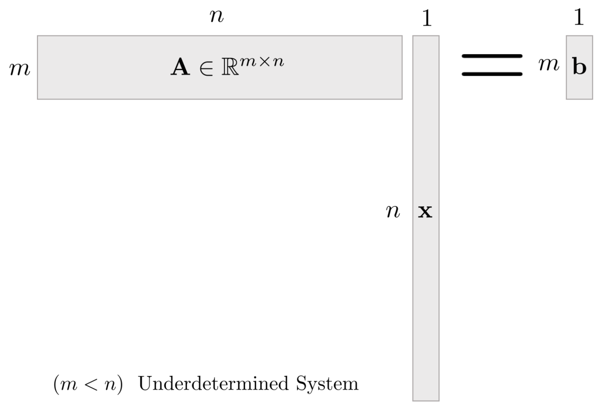


$(m > n)$ Overdetermined System

Over-determined 시스템은 방정식의 개수가 미지수의 개수보다 많은 경우를 의미한다. $\mathbf{A}\mathbf{x} = \mathbf{b}$ 의 형태에서 $\mathbf{A} \in \mathbb{R}^{m\times n}, \mathbf{x} \in \mathbb{R}^{n\times 1}, \mathbf{b} \in \mathbb{R}^{m\times 1}$ 이라고 했을 때 $m > n$ 인 경우를 의미한다. Over-determined 시스템의 경우 해가 존재하지 않으며 full column rank를 가진다.

$\mathbf{A}\mathbf{x} = \mathbf{b}$ 시스템의 해가 존재하지 않으므로 $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|$을 최소화하는 근사해를 구하는 방법을 주로 사용한다.

2.3 Under-determined system



$(m < n)$ Underdetermined System

Under-determined 시스템의 경우 방정식의 개수보다 미지수의 개수가 많은 경우를 의미한다. 즉, over-determined 시스템과 반대로 $n > m$ 인 경우를 의미한다. Under-determined 시스템의 경우 무수히 많은 해가 존재하며 full row rank를 가진다.

$\mathbf{Ax} = \mathbf{b}$ 시스템이 무수히 많은 해를 가지므로 $\|\mathbf{x}\|^2$ 가 최소가 되는 해를 구하는 방법을 주로 사용한다.

3 LU decomposition

LU 분해는 $\mathbf{Ax} = \mathbf{b}$ 의 시스템에서 행렬 $\mathbf{A}$를 하삼각(lower-triangle) 행렬 $\mathbf{L}$ 과 상삼각(upper-triangle) 행렬 $\mathbf{U}$의 곱으로 분해하는 방법이다.

$$\mathbf{A} = \mathbf{LU} = \mathbf{b} \tag{4}$$

LU 분해를 사용하면 다음과 같이 방정식이 변형된다.

$$\mathbf{Ax} = (\mathbf{LU})\mathbf{x} \tag{5}$$

이를 사용하여 [1] $\mathbf{Ly} = \mathbf{b}$ 방정식을 먼저 푼 후, [2] $\mathbf{Ux} = \mathbf{y}$ 를 순차적으로 계산할 수 있다. tridiagonal, band-diagonal 시스템에 효과적으로 사용할 수 있다.

$$\begin{aligned} &\mathbf{L}(\mathbf{Ux}) = \mathbf{b} \\ &\mathbf{Ly} = \mathbf{b} \quad \cdots [1] \\ &\mathbf{Ux} = \mathbf{y} \quad \cdots [2] \end{aligned} \tag{6}$$

위와 같이 행렬 $\mathbf{A}$를 $\mathbf{L}, \mathbf{U}$로 분해하면 $\mathbf{x}$를 두 스텝에 걸쳐 구해야하지만 삼각행렬의 특성 상 $\mathbf{x}$를 구하는 것이 훨씬 더 간단해진다.

3.1 PLU decomposition

만약 $\mathbf{A}$가 아래와 같은 3x3 행렬이라고 가정해보자.

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1^\intercal \\ \mathbf{a}_2^\intercal \\ \mathbf{a}_3^\intercal \end{bmatrix} = \begin{bmatrix} 0 & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \tag{7}$$

LU 분해는 가우스 조던 소거법으로 $\mathbf{L}$을 구하기 때문에 만약 $\mathbf{A}$의 첫 번째 원소가 0으로 시작하는 경우 정상적으로 분해할 수 없다. 따라서 첫번째 행과 두번째 행의 순서를 변환하는 permutation 행렬 $\mathbf{P}$를 앞에 곱해줘야 LU 분해를 수행할 수 있다.

$$\mathbf{PA} = \begin{bmatrix} & 1 & \\ 1 & & \\ & & 1 \end{bmatrix} \mathbf{A} = \begin{bmatrix} \mathbf{a}_2^\intercal \\ \mathbf{a}_1^\intercal \\ \mathbf{a}_3^\intercal \end{bmatrix} = \begin{bmatrix} * & * & * \\ 0 & * & * \\ * & * & * \end{bmatrix} \tag{8}$$

permutation 행렬은 직교행렬이고 직교행렬의 특성 상 $\mathbf{P} = \mathbf{P}^\intercal = \mathbf{P}^{-1}$이므로 이를 넘긴 후 전개하면 다음과 같다.

$$\mathbf{A} = \mathbf{PLU} \tag{9}$$

이와 같이 행벡터의 순서를 변경하는 $\mathbf{P}$를 곱한 후 LU 분해를 수행하는 방법을 PLU 분해라고 한다.

3.2 LDU decomposition

LU 분해에서 $\mathbf{L}, \mathbf{D}$ 행렬의 대각 성분을 1로 만들기 위해 중앙에 대각행렬 $\mathbf{D}$를 별도로 분해하는 방법을 LDU 분해라고 한다. 따라서 모든 LU 행렬은 LDU 행렬로 분해할 수 있다.

$$\mathbf{A} = \mathbf{LU} = \mathbf{L}'\mathbf{DU}' \tag{10}$$

4 Cholesky decomposition

Cholesky 분해는 $\mathbf{Ax} = \mathbf{b}$ 시스템에서 **$\mathbf{A}$가 대칭행렬이면서 동시에 postive(-semi) definite인 경우에 이를 하삼각(lower-triangle) 행렬 $\mathbf{L}$의 곱으로 분해하는 방법을 말한다.**

$$\mathbf{A} = \mathbf{LL}^\intercal \tag{11}$$

Cholesky 분해는 수치적으로 안정하다는 특징이 있다.

4.1 Detailed explanation

임의의 3x3 대칭행렬 $\mathbf{A}$가 주어졌다고 하자. 이를 cholesky 분해하면 다음과 같다.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{21} & a_{22} & a_{32} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \mathbf{LL}^\intercal = \begin{bmatrix} l_{11} & & \\ l_{21} & l_{22} & \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ & l_{22} & l_{32} \\ & & l_{33} \end{bmatrix} \tag{12}$$

$\mathbf{LL}^\intercal$을 자세히 전개하면 다음과 같다.

$$\begin{bmatrix} l_{11} & & \\ l_{21} & l_{22} & \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ & l_{22} & l_{32} \\ & & l_{33} \end{bmatrix} = \begin{bmatrix} l_{11}^2 & l_{21}l_{11} & l_{31}l_{11} \\ l_{21}l_{11} & l_{21}^2 + l_{22}^2 & l_{31}l_{21} + l_{32}l_{22} \\ l_{31}l_{11} & l_{31}l_{21} + l_{32}l_{22} & l_{31}^2 + l_{32}^2 + l_{33}^3 \end{bmatrix} \tag{13}$$

이를 통해 일대일로 비교하면 $\mathbf{L}$의 원소는 다음과 같이 구할 수 있다.

$$\begin{aligned} l_{11} &= \sqrt{a_{11}} \quad \cdots \text{ up to sign} \\ l_{21} &= a_{21}/l_{11} \\ l_{31} &= a_{31}/l_{11} \\ l_{21} &= \sqrt{a_{22} - l_{21}^2} \\ l_{32} &= (a_{32} - l_{31}l_{21})/l_{22} \\ l_{33} &= \sqrt{a_{33} - l_{31}^2 - l_{32}^2} \end{aligned} \tag{14}$$

이를 임의의 행렬에 대해 일반화하여 표현하면 다음과 같다.

$$\begin{aligned} l_{ii} &= \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2} \\ l_{ij} &= \frac{1}{l_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} \right) \end{aligned} \tag{15}$$

4.2 LDLT decomposition

Cholesky 분해에서 $\mathbf{L}$ 행렬의 대각 성분을 1로 만들기 위해 중앙에 대각행렬 $\mathbf{D}$를 별도로 분해하는 방법을 LDLT 분해라고 한다. 따라서 모든 cholesky 행렬은 LDLT 행렬로 분해할 수 있다.

$$\mathbf{A} = \mathbf{LL}^\intercal = \mathbf{L}'\mathbf{DL}'^\intercal \tag{16}$$

5 QR decomposition

QR 분해는 $\mathbf{Ax} = \mathbf{b}$ 시스템에서 행렬 $\mathbf{A}$를 직교(orthogonal) 행렬 $\mathbf{Q}$와 상삼각(upper-triangle) 행렬 $\mathbf{R}$의 곱으로 분해하는 방법을 말한다.

$$\mathbf{A} = \mathbf{QR} \tag{17}$$

$\mathbf{Q}$ 가 직교 행렬이므로 $\mathbf{QQ}^\intercal = \mathbf{I}$의 성질을 지닌다. 일반적으로 QR 분해는 LU 분해보다 느리지만 최소제곱법(least squares) 문제를 풀 때 효율적이어서 자주 사용된다.

5.1 Detailed explanation

임의의 3x3 행렬 $\mathbf{A}$가 주어졌을 때 이를 열벡터로 표현하면 다음과 같다. 자세한 내용은 해당 강의를 참고하면 된다.

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3] \tag{18}$$

해당 행렬에 gram-schmidt 직교화를 수행하면 임의의 직교행렬 $\mathbf{Q}$를 만들 수 있다.

$$\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3] \tag{19}$$

이 때, gram-schmidt 직교화의 특성 상 $\mathbf{q}_1$은 첫번째 열벡터와 동일한 단위 벡터이고 $\mathbf{q}_2$는 $\mathbf{q}_1$와 직교한 단위벡터이며 $\mathbf{q}_3$는 $\mathbf{q}_1, \mathbf{q}_2$와 직교한 단위벡터이다. 이를 통해 $\mathbf{a}_i$를 구할 수 있다.

$$\begin{aligned}
\mathbf{a}_1 &= \mathbf{a}_1^\intercal \mathbf{q}_1 \cdot \mathbf{q}_1 \\
\mathbf{a}_2 &= \mathbf{a}_2^\intercal \mathbf{q}_1 \cdot \mathbf{q}_1 + \mathbf{a}_2^\intercal \mathbf{q}_2 \cdot \mathbf{q}_2 \\
\mathbf{a}_3 &= \mathbf{a}_3^\intercal \mathbf{q}_1 \cdot \mathbf{q}_1 + \mathbf{a}_3^\intercal \mathbf{q}_2 \cdot \mathbf{q}_2 + \mathbf{a}_3^\intercal \mathbf{q}_3 \cdot \mathbf{q}_3
\end{aligned} \tag{20}$$

이를 행렬 형태로 표현하면 다음과 같은 상삼각 $\mathbf{R}$ 행렬을 얻을 수 있다.

$$\begin{aligned}
[\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3] &= [\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3] \begin{bmatrix} \mathbf{a}_1^\intercal \mathbf{q}_1 & \mathbf{a}_2^\intercal \mathbf{q}_1 & \mathbf{a}_3^\intercal \mathbf{q}_1 \\ & \mathbf{a}_2^\intercal \mathbf{q}_2 & \mathbf{a}_3^\intercal \mathbf{q}_2 \\ & & \mathbf{a}_3^\intercal \mathbf{q}_3 \end{bmatrix} \\
\mathbf{A} &= \mathbf{QR} \\
\mathbb{R}^{3\times3} &= \mathbb{R}^{3\times3}\mathbb{R}^{3\times3}
\end{aligned} \tag{21}$$

임의의 직사각 행렬에 대해서도 QR 분해를 수행할 수 있다. 만약 5x3 행렬 $\mathbf{A}$가 주어졌을 때 이는 다음과 같이 QR 분해된다.

$$\begin{aligned}
[\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3] &= [\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4, \mathbf{q}_5] \begin{bmatrix} \mathbf{a}_1^\intercal \mathbf{q}_1 & \mathbf{a}_2^\intercal \mathbf{q}_1 & \mathbf{a}_3^\intercal \mathbf{q}_1 \\ & \mathbf{a}_2^\intercal \mathbf{q}_2 & \mathbf{a}_3^\intercal \mathbf{q}_2 \\ & & \mathbf{a}_3^\intercal \mathbf{q}_3 \\ & & \\ & & \end{bmatrix} \\
\mathbf{A} &= \mathbf{QR} \\
\mathbb{R}^{5\times3} &= \mathbb{R}^{5\times5}\mathbb{R}^{5\times3}
\end{aligned} \tag{22}$$

$\mathbf{q}_4, \mathbf{q}_5$ 벡터는 곱셈에 의해 0이 되어서 실제 $\mathbf{A}$ 행렬에는 관여하지 않는다.

5.2 QR decomposition on least squares problem

Over-determined 시스템 $\mathbf{Ax} = \mathbf{b}$가 주어졌을 때 이에 대한 최적해는 다음과 같이 최소제곱법을 통해 구할 수 있다.

$$\begin{aligned}
&\min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\|_2^2 \\
&\mathbf{x} = (\mathbf{A}^\intercal \mathbf{A})^{-1} \mathbf{A}^\intercal \mathbf{b}
\end{aligned} \tag{23}$$

최소제곱법을 QR 분해를 통해 해석해보자. 임의의 직사각 행렬 $\mathbf{A}$를 QR 분해해보면 다음과 같다.

$$\begin{aligned}
\mathbf{A} &= \mathbf{QR} \\
&= \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix}
\end{aligned} \tag{24}$$

$\|\mathbf{Ax} - \mathbf{b}\|_2^2$를 QR 분해해보면 다음과 같다.

$$\begin{aligned}
\|\mathbf{Ax} - \mathbf{b}\|_2^2 &= \|\mathbf{QRx} - \mathbf{b}\|_2^2 \\
&= \|\mathbf{Q}(\mathbf{Rx} - \mathbf{Q}^\intercal \mathbf{b})\|_2^2 \\
&= \|\mathbf{Rx} - \mathbf{Q}^\intercal \mathbf{b}\|_2^2 \\
&= \left\| \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{Q}_1^\intercal \\ \mathbf{Q}_2^\intercal \end{bmatrix} \mathbf{b} \right\|_2^2
\end{aligned} \tag{25}$$

위 식에서 세번째 줄은 $\|\mathbf{Q}(\cdot)\|_2^2 = (\cdot)^\intercal\mathbf{Q}^\intercal\mathbf{Q}(\cdot) = (\cdot)^\intercal(\cdot) = \|(\cdot)\|_2^2$을 통해 구할 수 있다. 네번째 줄은 **QR** 행렬을 블록 행렬 $\mathbf{Q}_i, \mathbf{R}_i$로 표현한 모습이다. 벡터의 제곱의 합은 선형성을 가지므로 위 식의 마지막 줄을 전개하면 다음과 같다.

$$\left\| \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{Q}_1^\intercal \\ \mathbf{Q}_2^\intercal \end{bmatrix} \mathbf{b} \right\|_2^2 = \|\mathbf{R}_1\mathbf{x} - \mathbf{Q}_1^\intercal\mathbf{b}\|_2^2 + \|-\mathbf{Q}_2^\intercal\mathbf{b}\|_2^2 \tag{26}$$

따라서 $\min_{\mathbf{x}} \left(\|\mathbf{R}_1\mathbf{x} - \mathbf{Q}_1^\intercal\mathbf{b}\|_2^2 + \|-\mathbf{Q}_2^\intercal\mathbf{b}\|_2^2 \right)$를 최소화하는 $\mathbf{x}$는 다음과 같이 구할 수 있다.

$$\mathbf{x} = \mathbf{R}^{-1}\mathbf{Q}^\intercal\mathbf{b} \tag{27}$$

이 때, 최소제곱법 식의 크기는 $\|-\mathbf{Q}_2^\intercal\mathbf{b}\|_2^2$이다.

6 Eigen decomposition

정방행렬 $\mathbf{A} \in \mathbb{R}^{n\times n}$가 대각화 가능한 경우 다음과 같은 두 행렬 $\mathbf{V} \in \mathbb{R}^{n\times n}$과 대각행렬 $\mathbf{D} \in \mathbb{R}^{n\times n}$에 대하여 $\mathbf{A}$를 다음과 같이 분해할 수 있다.

$$\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1} \tag{28}$$

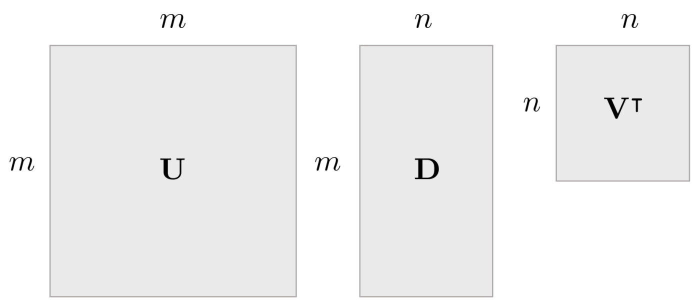
이를 행렬 A에 대한 고유값 분해(eigen decomposition) 라고 한다. 행렬 A가 대각화 가능하다는 의미는 행렬 A가 고유값 분해 가능하다는 말과 동치이다.

행렬 $\mathbf{A}$가 대각화(고유값분해)되기 위해서는 역행렬이 존재하는 행렬 $\mathbf{V}$가 존재해야 한다. 행렬 $\mathbf{V}$가 역행렬이 존재하기 위해서는 $\mathbf{V}$는 행렬 $\mathbf{A}$와 같은 $\mathbb{R}^{n\times n}$ 크기의 정방행렬이어야 하고 n개의 선형독립인 열벡터를 가지고 있어야 한다. 이 때, $\mathbf{V}$의 각 열은 행렬 $\mathbf{A}$의 고유벡터가 된다. 만약 행렬 $\mathbf{V}$가 존재하는 경우 행렬 $\mathbf{A}$는 대각화(고유값분해) 가능하다 고 한다.

7 Single value decomposition

$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\intercal =$

$(\mathbf{A} \in \mathbb{R}^{m\times n})$



행렬 $\mathbf{A} \in \mathbb{R}^{n\times m}$이 주어졌을 때 특이값 분해(Singular Value Decomposition, SVD)는 다음과 같이 나타낼 수 있다.

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\intercal \tag{29}$$

이 때, $\mathbf{U} \in \mathbb{R}^{m\times m}, \mathbf{V} \in \mathbb{R}^{n\times n}$인 행렬이며 이들은 각 열이 Col A와 Row A에 의 정규직교기저벡터(Orthonormal Basis)로 구성되어 있다. $\Sigma \in \mathbb{R}^{m\times n}$은 대각행렬이며 대각 성분들이 $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(m,n)}$ 특이값이며 큰 값부터 내림차순으로 정렬된 행렬이다.

7.1 Computing SVD

행렬 $\mathbf{A} \in \mathbb{R}^{m\times n}$에 대하여 $\mathbf{A}\mathbf{A}^\intercal$와 $\mathbf{A}^\intercal\mathbf{A}$는 다음과 같이 고유값분해할 수 있다.

$$\begin{aligned} \mathbf{A}\mathbf{A}^\intercal &= \mathbf{U}\Sigma\mathbf{V}^\intercal\mathbf{V}\Sigma^\intercal\mathbf{U}^\intercal = \mathbf{U}\Sigma\Sigma^\intercal\mathbf{U}^\intercal = \mathbf{U}\Sigma^2\mathbf{U}^\intercal \\ \mathbf{A}^\intercal\mathbf{A} &= \mathbf{V}\Sigma^\intercal\mathbf{U}^\intercal\mathbf{U}\Sigma\mathbf{V}^\intercal = \mathbf{V}\Sigma^\intercal\Sigma\mathbf{U}^\intercal = \mathbf{V}\Sigma^2\mathbf{V}^\intercal \end{aligned} \tag{30}$$

이 때 계산되는 행렬 $\mathbf{U}, \mathbf{V}$은 직교하는 고유벡터를 각 열의 성분으로 하는 행렬이며 대각행렬 Σ^2의 각 성분은 항상 0보다 큰 양수의 값을 가진다. 그리고 $\mathbf{A}\mathbf{A}^\intercal$와 $\mathbf{A}^\intercal\mathbf{A}$를 통해 계산되는 Σ^2의 값은 동일하다.

또한, 행렬 $\mathbf{A}$에 대하여 $\mathbf{A}\mathbf{A}^\intercal = \mathbf{A}^\intercal\mathbf{A} = \mathbf{S}$인 대칭행렬이 존재할 때 $\mathbf{S}$가 Positive (Semi-)Definite한 경우

$$\begin{aligned}\mathbf{x}^\intercal\mathbf{A}\mathbf{A}^\intercal\mathbf{x} &= (\mathbf{A}^\intercal\mathbf{x})^\intercal(\mathbf{A}^\intercal\mathbf{x}) = \|\mathbf{A}^\intercal\mathbf{x}\| \geq 0 \\ \mathbf{x}^\intercal\mathbf{A}^\intercal\mathbf{A}\mathbf{x} &= (\mathbf{A}\mathbf{x})^\intercal(\mathbf{A}\mathbf{x}) = \|\mathbf{A}\mathbf{x}\|^2 \geq 0\end{aligned} \tag{31}$$

즉, $\mathbf{A}\mathbf{A}^\intercal = \mathbf{U}\Sigma^2\mathbf{U}^\intercal$와 $\mathbf{A}^\intercal\mathbf{A} = \mathbf{V}\Sigma^2\mathbf{V}^\intercal$에서 Σ^2의 값은 항상 양수가 된다.

임의의 직각 행렬 $\mathbf{A} \in \mathbb{R}^{m\times n}$에 대하여 특이값 분해는 언제나 존재한다. $\mathbf{A} \in \mathbb{R}^{n\times n}$ 행렬의 경우 고유값 분해가 존재하지 않을 수 있지만 특이값 분해는 항상 존재한다. 대칭이면서 동시에 Positive Definite인 정방 행렬 $\mathbf{S} \in \mathbb{R}^{n\times n}$은 항상 고유값 분해값이 존재하며 이는 특이값 분해와 동일하다.

7.2 Range and nullspace of SVD

SVD는 다른 행렬 분해 방법들과 달리 A 가 Singular하거나 Near-Singular한 경우에도 사용할 수 있는 방법이다. $\mathbf{A}$ 가 Non-Singular한 경우 역행렬은 $\mathbf{A}^{-1} = \mathbf{V}\cdot\text{diag}(1/\sigma_j)\cdot\mathbf{U}^\intercal$ 와 같이 계산할 수 있다. 만약 $\mathbf{A}$ 가 Singular한 경우 역행렬을 구할 때 몇몇 $\sigma_j = 0$ 이 되는데 이 때 $1/\sigma_j \Rightarrow 0$ 으로 설정함으로써 역행렬을 구할 수 있다. 특이값 σ_j와 관련하여 SVD 분해는 다음과 같은 성질을 갖는다.

- $\sigma_j \neq 0$ 일 때 이와 상응하는 $\mathbf{U}$ 의 column들을 $\mathbf{A}$ 행렬의 Orthogonal set of basis vector of Range 라고 한다.
- $\sigma_j = 0$ 일 때 이와 상응하는 $\mathbf{V}$ 의 column들을 $\mathbf{A}$ 의 Orthogonal set of basis vector of Null Space 라고 한다.
- 0이 아닌 특이값 $\sigma_j \neq 0$의 개수는 곧 행렬 $\mathbf{A}$의 rank와 같다.

7.3 SVD on under-determined system

$\mathbf{A}$ 가 Singular이면서 동시에 $\mathbf{b}$ 가 Range 안에 포함되는 경우 선형시스템은 다수의 해를 가진다. 이런 경우 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 에서 $\|\mathbf{x}\|^2$ 가 최소가 되는 해를 구할 수 있다.

$$\begin{aligned}&\min_{\mathbf{x}} \|\mathbf{x}\|^2 \\ &\mathbf{x} = \mathbf{V}\cdot\text{diag}(1/\sigma_j)\cdot\mathbf{U}^\intercal\cdot\mathbf{b}\end{aligned} \tag{32}$$

7.4 SVD on over-determined system

$\mathbf{A}$ 가 Singular이면서 동시에 $\mathbf{b}$ 가 Range에 존재하지 않는 경우 선형시스템은 해가 존재하지 않는다. 이런 경우 $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|$ 가 최소가 되는 근사해를 구할 수 있다.

$$\begin{aligned}&\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\| \\ &\mathbf{x} = \mathbf{V}\cdot\text{diag}(1/\sigma_j)\cdot\mathbf{U}^\intercal\cdot\mathbf{b}\end{aligned} \tag{33}$$

8 Pseudo inverse

Pseudo Inverse는 선형시스템에서 행렬 $\mathbf{A}$가 정방행렬이 아닐 경우 임의로 역행렬을 구하는 방법을 말한다. **이 때, 선형시스템은 일반적으로 full column rank 또는 full row rank일 때 pseudo inverse를 적용할 수 있다.** Full rank가 아닌 행렬에 대한 pseudo inverse는 추후 섹션에서 설명한다.

8.1 Pseudo inverse on under-determined system

under-determined 시스템의 경우 $\mathbf{A}$는 full row rank가 되고 pseudo inverse는 다음과 같이 정의된다. lagrange multiplier λ를 포함하여 최적화 문제를 정의하면 다음과 같다.

$$\min_{\mathbf{x}} \|\mathbf{x}\|^2 + \lambda^\intercal(\mathbf{b} - \mathbf{A}\mathbf{x}) \tag{34}$$

이를 미분 후 0으로 만드는 값을 찾으면 다음과 같다.

$$2\mathbf{x} - \mathbf{A}^\intercal\lambda = 0 \tag{35}$$

$\mathbf{A}$가 정방행렬이 아니기 때문에 바로 해를 구할 수 없으므로 양변의 왼쪽에 $\mathbf{A}$를 곱한다.

$$2\mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{A}^\intercal\lambda = 0 \tag{36}$$

이 때, $\mathbf{A}\mathbf{x} = \mathbf{b}$이므로 다음과 같은 식이 유도된다.

$$2\mathbf{b} = \mathbf{A}\mathbf{A}^\intercal\lambda \tag{37}$$

$$\lambda = 2(\mathbf{A}\mathbf{A}^\intercal)^{-1}\mathbf{b} \tag{38}$$

따라서 $\mathbf{x}$는 다음과 같이 구할 수 있다.

$$\mathbf{x} = \mathbf{A}^\intercal(\mathbf{A}\mathbf{A}^\intercal)^{-1}\mathbf{b} \tag{39}$$

$$\boxed{\mathbf{A}^\dagger = \mathbf{A}^\intercal(\mathbf{A}\mathbf{A}^\intercal)^{-1} \quad \cdots \text{ for under-determined system}} \tag{40}$$

즉, under-determined 시스템에서 pseudo inverse는 오른쪽에 곱해지게 되며 이를 right pseudo inverse라고 부르기도 한다.

$$\begin{aligned}
&\mathbf{A}\mathbf{A}^\dagger\mathbf{x} = \mathbf{A}^\dagger\mathbf{b} \\
&\mathbf{x} = \mathbf{A}^\dagger\mathbf{b} \\
&\mathbf{x} = \mathbf{A}^\intercal(\mathbf{A}\mathbf{A}^\intercal)^{-1}\mathbf{b} \quad \cdots \text{ for under-determined system}
\end{aligned} \tag{41}$$

8.2 Pseudo inverse on over-determined system

over-determined 시스템의 경우 $\mathbf{A}$는 full column rank가 되고 pseudo inverse는 다음과 같이 정의된다. 우선, over-determined 시스템에 대한 최적화 문제는 다음과 같이 최소제곱법 문제가 된다.

$$\min_{\mathbf{x}} ||\mathbf{A}\mathbf{x} - \mathbf{b}||^2 = \min_{\mathbf{x}} ||\mathbf{b} - \mathbf{A}\mathbf{x}||^2 = \min_{\mathbf{x}} (\mathbf{b} - \mathbf{A}\mathbf{x})^\intercal(\mathbf{b} - \mathbf{A}\mathbf{x}) \tag{42}$$

이를 전개하면 다음과 같다.

$$\min_{\mathbf{x}} \mathbf{b}^\intercal\mathbf{b} - \mathbf{b}^\intercal\mathbf{A}\mathbf{x} - \mathbf{x}^\intercal\mathbf{A}^\intercal\mathbf{b} + \mathbf{x}^\intercal\mathbf{A}^\intercal\mathbf{A}\mathbf{x} \tag{43}$$

위 문제를 풀기 위해 $\mathbf{x}$에 대해 미분하면 다음과 같은 식이 얻어진다.

$$-(\mathbf{b}^\intercal\mathbf{A})^\intercal - (\mathbf{A}^\intercal\mathbf{b}) + 2\mathbf{A}^\intercal\mathbf{A}\mathbf{x} = 0 \tag{44}$$

따라서 $\mathbf{x}$는 다음과 같이 구할 수 있다.

$$\mathbf{x} = (\mathbf{A}^\intercal\mathbf{A})^{-1}\mathbf{A}^\intercal\mathbf{b} \tag{45}$$

$$\boxed{\mathbf{A}^\dagger = (\mathbf{A}^\intercal\mathbf{A})^{-1}\mathbf{A}^\intercal \quad \cdots \text{ for over-determined system}} \tag{46}$$

즉, over-determined 시스템에서 pseudo inverse는 왼쪽에 곱해지게 되며 이를 left pseudo inverse라고 부르기도 한다.

$$\begin{aligned}
&\mathbf{A}^\dagger\mathbf{A}\mathbf{x} = \mathbf{A}^\dagger\mathbf{b} \\
&\mathbf{x} = \mathbf{A}^\dagger\mathbf{b} \\
&\mathbf{x} = (\mathbf{A}^\intercal\mathbf{A})^{-1}\mathbf{A}^\intercal\mathbf{b} \quad \cdots \text{ for over-determined system}
\end{aligned} \tag{47}$$

8.3 SVD of pseudo inverse

선형 시스템 $\mathbf{A}\mathbf{x} = \mathbf{b}$이 주어졌을 때 임의의 직사각형 행렬 $\mathbf{A} \in \mathbb{R}^{m\times n}$와 pseudo inverse $\mathbf{A}^\dagger$는 SVD 분해를 통해 다음과 같이 나타낼 수 있다.

$$\begin{aligned}
\mathbf{A} &= \mathbf{U}\Sigma\mathbf{V}^\intercal \\
\mathbf{A}^\dagger &= \mathbf{V}\Sigma^\dagger\mathbf{U}^\intercal
\end{aligned} \tag{48}$$

- $\mathbf{U} \in \mathbb{R}^{m\times m}$
- $\Sigma = \text{diag}(\sigma_1, \sigma_2, \sigma_3, \cdots) \in \mathbb{R}^{m\times n}$
- $\Sigma^\dagger = \text{diag}(1/\sigma_1, 1/\sigma_2, 1/\sigma_3, \cdots) \in \mathbb{R}^{n\times m}$
- $\mathbf{V} \in \mathbb{R}^{n\times n}$

8.3.1 Full column rank case

임의의 직사각형 행렬 $\mathbf{A} \in \mathbb{R}^{m \times n}$가 full column rank를 가지면 pseudo inverse는 $\mathbf{A}^\dagger\mathbf{A}$와 같이 왼쪽에 곱해진다. 이를 SVD 분해하여 표현하면 다음과 같다.

$$\mathbf{A}^\dagger\mathbf{A} = \mathbf{V}\Sigma^\dagger\mathbf{U}^\intercal\mathbf{U}\Sigma\mathbf{V}^\intercal = \mathbf{I}_n \tag{49}$$

또한, 앞서 구한 (46)에서 $\mathbf{A}^\dagger = (\mathbf{A}^\intercal\mathbf{A})^{-1}\mathbf{A}^\intercal$를 풀어쓰면 다음과 같다.

$$\begin{aligned} \mathbf{A}^\dagger &= (\mathbf{A}^\intercal\mathbf{A})^{-1}\mathbf{A}^\intercal \\ &= (\mathbf{V}\Sigma^\intercal\mathbf{U}^\intercal\mathbf{U}\Sigma\mathbf{V}^\intercal)^{-1}\mathbf{V}\Sigma^\intercal\mathbf{U}^\intercal \\ &= \mathbf{V}\Sigma^{-2}\Sigma\mathbf{U}^\intercal & \cdots \Sigma^\intercal = \Sigma \\ &= \mathbf{V}\Sigma^\dagger\mathbf{U}^\intercal & \cdots \Sigma^\dagger = \Sigma^{-1} \end{aligned} \tag{50}$$

이는 앞서 정의한 (48)와 동일하다.

8.3.2 Full row rank case

임의의 직사각형 행렬 $\mathbf{A} \in \mathbb{R}^{m \times n}$가 full row rank를 가지면 pseudo inverse는 $\mathbf{A}\mathbf{A}^\dagger$와 같이 오른쪽에 곱해진다. 이를 SVD 분해하여 표현하면 다음과 같다.

$$\mathbf{A}\mathbf{A}^\dagger = \mathbf{U}\Sigma\mathbf{V}^\intercal\mathbf{V}\Sigma^\dagger\mathbf{U}^\intercal = \mathbf{I}_m \tag{51}$$

또한, 앞서 구한 (40)에서 $\mathbf{A}^\dagger = \mathbf{A}^\intercal(\mathbf{A}\mathbf{A}^\intercal)^{-1}$를 풀어쓰면 다음과 같다.

$$\begin{aligned} \mathbf{A}^\dagger &= \mathbf{A}^\intercal(\mathbf{A}\mathbf{A}^\intercal)^{-1} \\ &= \mathbf{V}\Sigma^\intercal\mathbf{U}^\intercal(\mathbf{U}\Sigma\mathbf{V}^\intercal\mathbf{V}\Sigma^\intercal\mathbf{U}^\intercal)^{-1} \\ &= \mathbf{V}\Sigma\Sigma^{-2}\mathbf{U}^\intercal & \cdots \Sigma^\intercal = \Sigma \\ &= \mathbf{V}\Sigma^\dagger\mathbf{U}^\intercal & \cdots \Sigma^\dagger = \Sigma^{-1} \end{aligned} \tag{52}$$

이는 앞서 정의한 (48)와 동일하다.

8.3.3 Rank deficient case

만약 임의의 직사각형 행렬 $\mathbf{A}$이 full rank가 아닐 경우 pseudo inverse는 다음과 같이 나타낼 수 있다. 예를 들어 $\mathbf{A} \in \mathbb{R}^{3\times 4}$ 일 때 이를 SVD 분해하면 다음과 같다.

$$\begin{aligned} \mathbf{A} &= \mathbf{U}\Sigma\mathbf{V}^\intercal \\ &= \mathbf{U}\begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\mathbf{V}^\intercal \end{aligned} \tag{53}$$

Pseudo inverse $\mathbf{A}^\dagger$를 SVD 분해하면 다음과 같다.

$$\begin{aligned} \mathbf{A}^\dagger &= \mathbf{V}\Sigma^\dagger\mathbf{U}^\intercal \\ &= \mathbf{V}\begin{bmatrix} 1/\sigma_1 & 0 & 0 \\ 0 & 1/\sigma_2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\mathbf{U}^\intercal \end{aligned} \tag{54}$$

따라서 오른쪽 pseudo inverse $\mathbf{A}\mathbf{A}^\dagger$는 다음과 같이 전개할 수 있다.

$$\begin{aligned} \mathbf{A}\mathbf{A}^\dagger &= \mathbf{U}\Sigma\mathbf{V}^\intercal\mathbf{V}\Sigma^\dagger\mathbf{U}^\intercal \\ &= \mathbf{U}\begin{bmatrix} 1 & & \\ & 1 & \\ & & \end{bmatrix}\mathbf{U}^\intercal \\ &= \mathbf{u}_1\mathbf{u}_1^\intercal + \mathbf{u}_2\mathbf{u}_2^\intercal \end{aligned} \tag{55}$$

만약 $\mathbf{A}$가 full rank인 경우 $\mathbf{A}\mathbf{A}^{\dagger}$는 다음과 같다.

$$\begin{aligned} \mathbf{A}\mathbf{A}^{\dagger} &= \mathbf{I}_3 \\ &= \mathbf{u}_1\mathbf{u}_1^{\intercal} + \mathbf{u}_2\mathbf{u}_2^{\intercal} + \mathbf{u}_3\mathbf{u}_3^{\intercal} \end{aligned} \tag{56}$$

따라서 rank deficient 케이스의 경우 $\mathbf{A}\mathbf{A}^{\dagger}$는 마지막 $\mathbf{u}_3\mathbf{u}_3^{\intercal}$이 없는 pseudo inverse가 구해진다. 이는 항등행렬 $\mathbf{I}_3$와 유사한 값을 갖지만 동일하지는 않은 행렬이다.

다음으로 왼쪽 pseudo inverse $\mathbf{A}^{\dagger}\mathbf{A}$는 다음과 같이 전개할 수 있다.

$$\begin{aligned} \mathbf{A}^{\dagger}\mathbf{A} &= \mathbf{V}\Sigma^{\dagger}\mathbf{U}^{\intercal}\mathbf{U}\Sigma\mathbf{V}^{\intercal} \\ &= \mathbf{V}\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & & \\ & & & \end{bmatrix}\mathbf{V}^{\intercal} \\ &= \mathbf{v}_1\mathbf{v}_1^{\intercal} + \mathbf{v}_2\mathbf{v}_2^{\intercal} \end{aligned} \tag{57}$$

만약 $\mathbf{A}$가 full rank인 경우 $\mathbf{A}^{\dagger}\mathbf{A}$는 다음과 같다.

$$\begin{aligned} \mathbf{A}^{\dagger}\mathbf{A} &= \mathbf{I}_4 \\ &= \mathbf{v}_1\mathbf{v}_1^{\intercal} + \mathbf{v}_2\mathbf{v}_2^{\intercal} + \mathbf{v}_3\mathbf{v}_3^{\intercal} + \mathbf{v}_4\mathbf{v}_4^{\intercal} \end{aligned} \tag{58}$$

따라서 rank deficient 케이스의 경우 $\mathbf{A}^{\dagger}\mathbf{A}$는 마지막 $\mathbf{v}_3\mathbf{v}_3^{\intercal} + \mathbf{v}_4\mathbf{v}_4^{\intercal}$이 없는 pseudo inverse가 구해진다. 이는 항등행렬 $\mathbf{I}_4$와 유사한 값을 갖지만 동일하지는 않은 행렬이다. 이는 $\mathbf{A}\mathbf{A}^{\dagger}$를 통해 구한 행렬보다 덜 항등행렬에 근접하다.

따라서 임의의 non-full rank를 가지는 직사각형 행렬 $\mathbf{A} \in \mathbb{R}^{m \times n}$이 주어졌을 때, $m < n$인 경우 $\mathbf{A}\mathbf{A}^{\dagger}$를 수행하는 것이 더 항등행렬에 근접한 pseudo inverse를 수행할 수 있고 $m > n$인 경우 $\mathbf{A}^{\dagger}\mathbf{A}$를 수행하는 것이 더 항등행렬에 근접한 pseudo inverse를 수행할 수 있다.

8.4 QR decomposition of pseudo inverse when singular case

간혹 $\mathbf{A}^{\intercal}\mathbf{A}$ 가 singular하거나 near-singular한 경우 QR 분해를 사용하여 pseudo inverse를 구한다.

$$\begin{aligned} \mathbf{x} &= \mathbf{A}^{\dagger}\mathbf{b} \\ &= (\mathbf{A}^{\intercal}\mathbf{A})^{-1}\mathbf{A}^{\intercal}\mathbf{b} \\ &= (\mathbf{R}^{\intercal}\mathbf{Q}^{\intercal}\mathbf{Q}\mathbf{R})^{-1}\mathbf{R}^{\intercal}\mathbf{Q}^{\intercal}\mathbf{b} \\ &= (\mathbf{R}^{\intercal}\mathbf{R})^{-1}\mathbf{R}^{\intercal}\mathbf{Q}^{\intercal}\mathbf{b} \\ &= \mathbf{R}^{-1}\mathbf{Q}^{\intercal}\mathbf{b} \end{aligned} \tag{59}$$

위 식은 (27)와 동일하다.

9 Sherman-Morrison formula

역행렬이 존재하는 임의의 행렬 $\mathbf{A}$에 대해 rank 1 업데이트를 하는 방법을 Sherman-Morrison 공식이라고 한다. 해당 공식에 대한 보다 자세한 내용은 해당 강의를 참조하면 된다.

$$\boxed{(\mathbf{A} + \mathbf{u}\mathbf{v}^{\intercal})^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^{\intercal}\mathbf{A}^{-1}}{1 + \mathbf{v}^{\intercal}\mathbf{A}^{-1}\mathbf{u}}} \tag{60}$$

위 식에서 $(1 + \mathbf{v}^{\intercal}\mathbf{A}^{-1}\mathbf{u}) \neq 0$와 $(\mathbf{A} + \mathbf{u}\mathbf{v}^{\intercal})^{-1}$이 역행렬이 존재하는 조건은 동치이다. 이 때, $\mathbf{u}, \mathbf{v}$는 임의의 두 벡터를 의미하며 이를 $\mathbf{u}\mathbf{v}^{\intercal}$와 같이 곱하면 항상 rank 1 행렬이 생성된다.

$$\mathbf{u}\mathbf{v}^{\intercal} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}\begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} v_1\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} & v_2\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \end{bmatrix} \quad \cdots \text{ linearly dependent} = \text{rank 1} \tag{61}$$

9.1 Recursive least squares

Sherman-Morrison 공식은 데이터가 계속 추가되는 최소제곱법 문제에 사용하면 연산량을 적게 소모하면서 효율적으로 역행렬을 업데이트할 수 있다. 다음과 같은 선형 시스템 $\mathbf{Ax} = \mathbf{b}$가 주어졌다고 하자. $\mathbf{A} \in \mathbb{R}^{m\times n}, \mathbf{x} \in \mathbb{R}^{n\times 1}, \mathbf{b} \in \mathbb{R}^{m\times 1}$ 일 때 이를 풀어서 쓰면 아래와 같다.

$$\begin{gathered}\mathbf{Ax} = \mathbf{b} \\ \begin{bmatrix} \mathbf{a}_1^\intercal \\ \mathbf{a}_2^\intercal \\ \vdots \\ \mathbf{a}_m^\intercal \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}\end{gathered} \tag{62}$$

선형시스템의 최소제곱법의 해는 다음과 같다.

$$\mathbf{x} = (\mathbf{A}^\intercal\mathbf{A})^{-1}\mathbf{A}^\intercal\mathbf{b} \tag{63}$$

만약 $m+1$ 번째 데이터 $\mathbf{a}_{m+1}^\intercal$이 입력되면 이에 맞게 최적해를 업데이트해줘야 한다. 표현의 편의를 위해 $m+1$ 번째 데이터를 $\mathbf{a}$로 표현하면 다음과 같다.

$$\begin{aligned}\mathbf{x} &= \left(\begin{bmatrix} \mathbf{A}^\intercal & \mathbf{a} \end{bmatrix} \begin{bmatrix} \mathbf{A}^\intercal \\ \mathbf{a} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{A}^\intercal & \mathbf{a} \end{bmatrix} \begin{bmatrix} \mathbf{b} \\ b \end{bmatrix} \\ &= (\mathbf{A}^\intercal\mathbf{A} + \mathbf{a}\mathbf{a}^\intercal)^{-1}(\mathbf{A}^\intercal\mathbf{b} + \mathbf{a}b_{m+1})\end{aligned} \tag{64}$$

이 때, 앞 부분 $(\mathbf{A}^\intercal\mathbf{A} + \mathbf{a}\mathbf{a}^\intercal)^{-1}$에 Sherman-Morisson 공식 (60)을 적용하면 연산량을 적게 소모하면서 효율적으로 최적해를 업데이트할 수 있다. 이는 다음과 같이 전개 후 치환하여 간결하게 나타낼 수 있다.

$$\begin{aligned}(\mathbf{A}^\intercal\mathbf{A} + \mathbf{a}\mathbf{a}^\intercal)^{-1} &= (\mathbf{A}^\intercal\mathbf{A})^{-1} - \frac{(\mathbf{A}^\intercal\mathbf{A})^{-1}\mathbf{a}\mathbf{a}^\intercal(\mathbf{A}^\intercal\mathbf{A})^{-1}}{1 + \mathbf{a}^\intercal(\mathbf{A}^\intercal\mathbf{A})^{-1}\mathbf{a}} \\ &= \mathbf{P} - \frac{\mathbf{P}\mathbf{a}\mathbf{a}^\intercal\mathbf{P}}{1 + \mathbf{a}^\intercal\mathbf{P}\mathbf{a}} \\ &= \mathbf{P}_a\end{aligned} \tag{65}$$

- $(\mathbf{A}^\intercal\mathbf{A})^{-1} = \mathbf{P}$ 표현의 편의를 위해 치환한다

위 치환한 식을 기반으로 (64)를 전개하면 다음과 같다.

$$\begin{aligned}(\mathbf{A}^\intercal\mathbf{A} + \mathbf{a}\mathbf{a}^\intercal)^{-1}(\mathbf{A}^\intercal\mathbf{b} + \mathbf{a}b_{m+1}) &= \left(\mathbf{P} - \frac{\mathbf{P}\mathbf{a}\mathbf{a}^\intercal\mathbf{P}}{1 + \mathbf{a}^\intercal\mathbf{P}\mathbf{a}}\right)(\mathbf{A}^\intercal\mathbf{b} + \mathbf{a}b_{m+1}) \\ &= \mathbf{P}\mathbf{A}^\intercal\mathbf{b} + \frac{\mathbf{P}\mathbf{a}\mathbf{a}^\intercal\mathbf{P}}{1 + \mathbf{a}^\intercal\mathbf{P}\mathbf{a}}\mathbf{A}^\intercal\mathbf{b} + \mathbf{P}_a\mathbf{a}b \\ &= \mathbf{x} - \frac{\mathbf{P}\mathbf{a}\mathbf{a}^\intercal\mathbf{P}}{1 + \mathbf{a}^\intercal\mathbf{P}\mathbf{a}}\mathbf{A}^\intercal\mathbf{b} + \mathbf{P}_a\mathbf{a}b \\ &= \mathbf{x} - \left(\frac{\mathbf{P}\mathbf{a}}{1 + \mathbf{a}^\intercal\mathbf{P}\mathbf{a}}\right)\mathbf{a}^\intercal\mathbf{x} + \mathbf{P}_a\mathbf{a}b \\ &= \mathbf{x} - (\mathbf{P}_a\mathbf{a})\mathbf{a}^\intercal\mathbf{x} + \mathbf{P}_a\mathbf{a}b \\ &= \mathbf{x} + \mathbf{P}_a\mathbf{a}(b - \mathbf{a}^\intercal\mathbf{x})\end{aligned} \tag{66}$$

- $\mathbf{P}\mathbf{A}^\intercal\mathbf{b} = \mathbf{x}$

위 식에서 5번째 줄은 $\mathbf{P}_a\mathbf{a}$를 전개한 후 분모를 통분하여 정리함으로써 유도할 수 있다. **따라서 데이터가 증가했을 때 새로운 최적해는 이전 최적해 식으로부터 아래와 같이 업데이트된다. 이를 recursive least squares(RLS)라고 한다.**

$$\boxed{\mathbf{x} \leftarrow \mathbf{x} + \mathbf{P}_a\mathbf{a}(b - \mathbf{a}^\intercal\mathbf{x})} \tag{67}$$

10 Matrix inversion lemma

Matrix inversion lemma는 역행렬 변환 공식을 의미하며 선형 시스템을 다룰 때 자주 쓰이는 트릭 중 하나이다. 이는 Sherman-Morrison-Woodbury 공식이라고도 불린다. Matrix inversion lemma는 는 다음과 같이

정의된다. Lemma에 대한 보다 자세한 내용은 해당 강의를 참조하면 된다.

$$\boxed{(\mathbf{A}+\mathbf{UCV})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1}+\mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1}} \tag{68}$$

- $\mathbf{A} \in \mathbb{R}^{n\times n}$
- $\mathbf{U} \in \mathbb{R}^{n\times k}$
- $\mathbf{C} \in \mathbb{R}^{k\times k}$
- $\mathbf{V} \in \mathbb{R}^{k\times n}$
- $\mathbf{A}, \mathbf{C}, \mathbf{C}^{-1}+\mathbf{VA}^{-1}\mathbf{U}$ is invertible

10.1 Derivation of matrix inversion lemma

Matrix inversion lemma를 유도하기 위해 4개의 블록 행렬로 구성된 $\mathbf{M}$가 주어졌다고 하자.

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \tag{69}$$

10.1.1 LDU decomposition

다음으로 $\mathbf{M}$를 LDU 분해하려고 한다. 아래와 같이 $\mathbf{C}$를 소거하기 위한 행렬을 곱해서 LU 행렬을 만들 수 있다.

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{CA}^{-1} & \mathbf{I} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{CA}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{CA}^{-1} & \mathbf{I} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{D}-\mathbf{CA}^{-1}\mathbf{B} \end{bmatrix} \tag{70}$$

이 때, $\mathbf{D}-\mathbf{CA}^{-1}\mathbf{B}$를 $\mathbf{A}$의 schur complement $(\mathbf{M}/\mathbf{A})$라고 한다. 다음으로 대각 행렬 성분만 남기기 위해 아래와 같이 오른쪽에 행렬을 전개하면 LDU 분해가 마무리된다.

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{CA}^{-1} & \mathbf{I} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{D}-\mathbf{CA}^{-1}\mathbf{B} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{CA}^{-1} & \mathbf{I} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}-\mathbf{CA}^{-1}\mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{71}$$

$\mathbf{M}^{-1}$은 다음과 같이 LDU 행렬을 사용하여 전개할 수 있다.

$$\boxed{\begin{aligned} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} &= \begin{bmatrix} \mathbf{I} & -\mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{0} \\ \mathbf{0} & (\mathbf{D}-\mathbf{CA}^{-1}\mathbf{B})^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{C}^{-1}\mathbf{A} & \mathbf{I} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A}^{-1}+\mathbf{A}^{-1}\mathbf{B}(\mathbf{D}-\mathbf{CA}^{-1}\mathbf{B})^{-1}\mathbf{CA}^{-1} & -\mathbf{A}^{-1}\mathbf{B}(\mathbf{D}-\mathbf{CA}^{-1}\mathbf{B})^{-1} \\ -(\mathbf{D}-\mathbf{CA}^{-1}\mathbf{B})^{-1}\mathbf{CA}^{-1} & (\mathbf{D}-\mathbf{CA}^{-1}\mathbf{B})^{-1} \end{bmatrix} \end{aligned}} \tag{72}$$

10.1.2 UDL decomposition

행렬 $\mathbf{M}$ LDU 뿐만아니라 UDL로도 분해될 수 있다. 아래와 같이 $\mathbf{B}$를 소거하기 위한 행렬을 곱해서 UL 행렬을 만들 수 있다.

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{D}^{-1}\mathbf{C} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{D}^{-1}\mathbf{C} & \mathbf{I} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}-\mathbf{BD}^{-1}\mathbf{C} & \mathbf{B} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{D}^{-1}\mathbf{C} & \mathbf{I} \end{bmatrix}^{-1} \tag{73}$$

이 때, $\mathbf{A}-\mathbf{BD}^{-1}\mathbf{C}$를 $\mathbf{D}$의 schur complement $(\mathbf{M}/\mathbf{D})$라고 한다. 다음으로 대각 행렬 성분만 남기기 위해 아래와 같이 왼쪽에 행렬을 전개하면 UDL 분해가 마무리된다.

$$\begin{bmatrix} \mathbf{A}-\mathbf{BD}^{-1}\mathbf{C} & \mathbf{B} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{D}^{-1}\mathbf{C} & \mathbf{I} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{I} & \mathbf{BD}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A}-\mathbf{BD}^{-1}\mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{D}^{-1}\mathbf{C} & \mathbf{I} \end{bmatrix}^{-1} \tag{74}$$

$\mathbf{M}^{-1}$은 다음과 같이 UDL 행렬을 사용하여 전개할 수 있다.

$$\boxed{\begin{aligned} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{D}^{-1}\mathbf{C} & \mathbf{I} \end{bmatrix} \begin{bmatrix} (\mathbf{A}-\mathbf{BD}^{-1}\mathbf{C})^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{BD}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \\ &= \begin{bmatrix} (\mathbf{A}-\mathbf{BD}^{-1}\mathbf{C})^{-1} & -(\mathbf{A}-\mathbf{BD}^{-1}\mathbf{C})^{-1}\mathbf{BD} \\ -\mathbf{D}^{-1}\mathbf{C}(\mathbf{A}-\mathbf{BD}^{-1}\mathbf{C})^{-1} & \mathbf{D}^{-1}+\mathbf{D}^{-1}\mathbf{C}(\mathbf{A}-\mathbf{BD}^{-1}\mathbf{C})^{-1}\mathbf{BD} \end{bmatrix} \end{aligned}} \tag{75}$$

10.1.3 Back to matrix inversion lemma

앞서 구한 (72), (75)는 분해 방법만 달랐을 뿐 모든 원소는 서로 같아야 한다. 따라서 첫번째 원소를 비교해 보면 다음과 같다.

$$(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} \tag{76}$$

해당 식에서 아래와 같이 기호만 변경해주면 matrix inversion lemma 식 (68)가 된다.

$$\begin{aligned} \mathbf{B} &\rightarrow \mathbf{U} \\ \mathbf{C} &\rightarrow \mathbf{V} \\ \mathbf{D}^{-1} &\rightarrow -\mathbf{C} \\ \therefore (\mathbf{A} + \mathbf{U}\mathbf{C}\mathbf{V})^{-1} &= \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1} + \mathbf{V}\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{A}^{-1} \end{aligned} \tag{77}$$

또한 (72), (75)의 두번째 원소를 비교하면 다음과 같다. 해당 식도 자주 사용되는 행렬 변환 트릭 중 하나이다.

$$\boxed{-\mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} = -(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}\mathbf{B}\mathbf{D}} \tag{78}$$

지금까지 소개한 matrix inversion lemma 행렬 변환 트릭은 칼만 필터(kalman filter)의 공식을 유도할 때 종종 사용되며 이외에도 많은 공학 분야에서 사용된다.

11 Reference

[1] [blog] 선형대수학 (Linear Algebra) 개념 정리

[2] [pdf] Pseudo Inverse 유도 과정

[3] 혁펜하임님 유튜브 강의 영상

12 Revision log

- 1st: 2020-06-02
- 2nd: 2023-01-21
- 3rd: 2023-01-23
- 4th: 2024-02-08