[선형대수학 #4] 특이값 분해(Singular Value Decomposition, SVD)의 활용

수학 이야기 2013. 10. 23. 09:43
활용도 측면에서 선형대수학의 꽃이라 할 수 있는 특이값 분해(Singular Value Decomposition, SVD)에 대한 내용입니다. 보통은 복소수 공간을 포함하여 정의하는 것이 일반적이지만 이 글에서는 실수(real) 공간에 한정하여 내용을 적겠습니다.

1. 특이값분해(Singular Value Decomposition, SVD)

특이값 분해(SVD)는 고유값 분해(eigendecomposition)처럼 행렬을 대각화하는 한 방법이다. 그런데, 특이값 분해가 유용한 이유는 행렬이 정방행렬이든 아니든 관계없이 모든 m x n 행렬에 대해 적용 가능하기 때문이다. [선형대수학 #3] 고유값과 고유벡터 (eigenvalue & eigenvector)에서 다루었던 고유값 분해(EVD)는 정방행렬에 대해서만 적용 가능하며 또한 정방행렬 중에서도 일부 행렬에 대해서만 적용 가능한 대각화 방법임을 상기하자.

실수공간에서 임의의 m x n 행렬에 대한 특이값분해(SVD)는 다음과 같이 정의된다.

$$A=U\Sigma V^T \quad \text{--- (1)}$$

$$U:\ m\times m\ \text{직교행렬}\ \ (AA^T=U(\Sigma\Sigma^T)U^T)$$
$$V:\ n\times n\ \text{직교행렬}\ \ (A^TA=V(\Sigma^T\Sigma)V^T)$$
$$\Sigma:\ m\times n\ \text{직사각 대각행렬}$$

U는 AA^T를 고유값분해(eigendecomposition)해서 얻어진 직교행렬(orthogonal matrix)로 U의 열벡터들을 A의 left singular vector라 부른다. 또한 V는 A^TA를 고유값분해해서 얻어진 직교행렬로서 V 의 열벡터들을 A의 right singular vector라 부른다. left, right가 상당히 햇갈리는데 그냥 Σ의 왼쪽에 있는 U가 left singular 벡터, 오른쪽에 있는 V가 right singular 벡터들이라고 생각하면 된다. 마지막으로, Σ는 AA^T, A^TA를 고유값분해해서 나오는 고유값(eigenvalue)들의 square root를 대각원소로 하는 m x n 직사각 대각행렬로 그 대각원소들을 A의 특이값(singular value)이라 부른다. U, V가 직교행렬(orthogonal matrix)이라 함은 $UU^T = VV^T = E$, $U^{-1}=U^T$, $V^{-1}=V^T$임도 기억하자.

U의 열벡터(ui): left singular vectors of A (AA^T의 eigenvector)
V의 열벡터(vi): right singular vectors of A (A^TA의 eigenvector)
Σ의 대각원소(σi): singular values of A (A^TA, AA^T의 eigenvalue들의 square root)

$$\Sigma=\begin{pmatrix}\sigma_1 & & \\ & \ddots & \\ & & \sigma_s \\ & & 0\end{pmatrix}\ (m>n)\ \ or\ \ \Sigma=\begin{pmatrix}\sigma_1 & & & \\ & \ddots & & \\ & & \sigma_s & 0\end{pmatrix}\ (m<n)$$

--- (2)

이전 글인 [선형대수학 #3] 고유값과 고유벡터 (eigenvalue & eigenvector)에서 설명했듯이 대칭행렬(symmetric matrix)은 항상 고유값 분해(eigendecomposition)가 가능하며 더구나 직교행렬(orthogonal matrix)로 대각화할 수 있다. 그런데, AA^T와 A^TA는 모두 대칭행렬(symmetric matrix)이므로 위와 같은 고유값 분해가 항상 가능하다.

여기서 재미있는 사실은 AA^T와 A^TA의 고유값들은 모두 0 이상(nonnegative)이며 0이 아닌 고유값들은 서로 동일하다는 점이다. 고유값들이 0 이상이어야 square root를 씌울수 있으며 또한 서로 동일해야만 하나의 행렬 Σ로 표현할 수 있을 것이다.

i) A^TA의 고유값을 λ, 고유벡터를 v (v≠0)라 했을 때, $A^TAv = \lambda v$ 이다. 양변에 v^T를 곱해보면 $v^TA^TAv = \lambda v^Tv$ 에서 $(Av)^TAv = \lambda v^Tv$ 즉, $\|Av\|^2 = \lambda\|v\|^2$ 이므로 λ≥0 이다.

ii) A^TA의 0이 아닌 고유값을 λ, 고유벡터를 v (v≠0)라 했을 때, 정의에 의해 $(A^TA)v=\lambda v$ 이다. 양변에 A를 곱해보면 $A(A^TA)v = \lambda Av$ 즉, $AA^T(Av) = \lambda(Av)$이므로 Av≠0라면 λ는 또한 AA^T의 고유값임을 알 수 있다. 그런데, 만일 Av=0라 하면 $(A^TA)v=\lambda v$에서 λ=0이어야 하므로 Av≠0이다. 동일한 방식으로 AA^T의 0이 아닌 모든 고유값도 또한 A^TA의 고유값이 됨을 쉽게 보일 수 있다.

이와같이, AA^T와 A^TA의 공통의 고유값 $\sigma_1^2\geq\sigma_2^2\geq \dots \geq\sigma_s^2\geq 0$ (단, s = min(m, n))들을 구한 후 이들의 square root를 취한 $\sigma_1\geq\sigma_2\geq \dots \geq\sigma_s\geq 0$ 이 A의 특이값(singular value)들이고 이들을 대각원소로 하는 m x n 직사각 대각행렬이 Σ이다 (식 (2)). 여기서 행렬의 특이값(singular value)은 모두 0 이상임을 기억하자.

참고로 행렬 A의 특이값(singular value)을 σi, left singular vector를 ui, right singular vector를 vi라 했을 때 다음 식이 성립한다.

$$Av_i=\sigma_i u_i,\ \ 1\leq i\leq s$$

--- (3)

※ 어떤 행렬 A가 singular하다는 말은 특이값 분해와는 조금 다른 말이다. 행렬 A가 singular하다는 말은 정방행렬(square matrix)에 대해서만 해당되는 말로 A의 역행렬이 존재하지 않는다는 말이다. 즉, det(A) = 0인 정방행렬을 특이행렬(singular matrix)이라고 부른다. 단, 정방행렬에 대해서도 특이값 분해를 하게 되면 그 행렬이 singular한지 안한지를 바로 알 수 있다. 정방행렬 A의 특이값들이 모두 0이 아니면 이 행렬은 nonsingular 즉, 역행렬이 존재하고 특이값중 0이 포함되면 singular 즉, 역행렬이 존재하지 않는다.

※ 특이값의 부호가 양수인 이유

앞서 singular value는 AA^T, A^TA의 고유값분해(EVD)에서 나온 고유값의 square root를 취한 것이라고 했다. 그런데, 고유값의 square root는 +, - 2개의 값이 나오는데 왜 +값만 singular value로 잡은 것일까? 사실 원래는 singular value를 +로 잡을수도, -로 잡을수도 있다. 하지만 +를 singular value로 잡는 것은 일종의 수학적 약속이다. 만일 식 (1)에서 i번째 singular value의 부호를 바꿨을 때 U의 i번째 열벡터의 부호를 같이 바꾸면 식 (1)은 여전히 성립함을 알 수 있다. 또한 U가 직교행렬(orthogonal matrix)란 점도 변하지 않는다. 어쨌든, 특이값(singular value)은 항상 0 이상(nonnegative)임을 기억하자.

※ positive definite, negative definite

선형대수학에서 보면 가끔 어떤 행렬 A가 positive definite다, positive semidefinite다 하는 용어들이 나오는데 이는

대칭행렬(symmetric matrix)에 대해서만 정의되는 용어들로서 이 용어들의 정의는 다음과 같다.

$$\forall z \neq 0,\ z^T A z > 0 \rightarrow positive\ definite$$
$$\forall z \neq 0,\ z^T A z \geq 0 \rightarrow positive\ semidefinite$$
$$\forall z \neq 0,\ z^T A z < 0 \rightarrow negative\ definite$$
$$\forall z \neq 0,\ z^T A z \leq 0 \rightarrow negative\ semidefinite$$

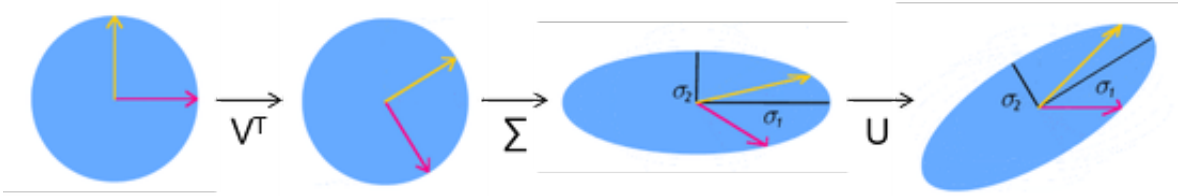
즉, 어떤 대칭행렬 A가 영벡터가 아닌 모든 열벡터 z에 대해 항상 $z^TAz>0$을 만족할 때 이 행렬을 positive definite 행렬이라고 부르며 negative definite 등도 유사하게 정의된다. 그런데 이러한 행렬들의 중요한 성질은 고유값(eigenvalue)도 동일한 부호를 갖는다는 점이다. 즉, 어떤 대칭행렬 A가 positive definite하면 이 행렬의 모든 eigenvalue은 항상 양수이다. 만일 A가 positive semidefinite하면 eigenvalue들도 ≥0이고, negative definite면 모든 eigenvalue들이 음수이다. 앞서, SVD를 구하는 과정에서 나왔던 AA^T, A^TA는 positive semidefinite 행렬들로서 항상 0 이상의 eigenvalue들을 갖는다.

2. 특이값분해(SVD)의 기하학적 의미

행렬을 x' = Ax와 같이 좌표공간에서의 선형변환으로 봤을 때 직교행렬(orthogonal matrix)의 기하학적 의미는 회전변환(rotation transformation) 또는 반전된(reflected) 회전변환, 대각행렬(diagonal maxtrix)의 기하학적 의미는 각 좌표성분으로의 스케일변환(scale transformation)이다.

행렬 R이 직교행렬(orthogonal matrix)이라면 $RR^T = E$이다. 따라서 $\det(RR^T) = \det(R)\det(R^T) = \det(R)^2 = 1$이므로 det(R)는 항상 +1, 또는 -1이다. 만일 det(R)=1라면 이 직교행렬은 회전변환을 나타내고 det(R)=-1라면 뒤집혀진(reflected) 회전변환을 나타낸다.

따라서 식 (1), $A = U\Sigma V^T$에서 U, V는 직교행렬, Σ는 대각행렬이므로 Ax는 x를 먼저 V^T에 의해 회전시킨 후 Σ로 스케일을 변화시키고 다시 U로 회전시키는 것임을 알 수 있다.



<그림1> 출처: 위키피디아

즉, 행렬의 특이값(singular value)이란 이 행렬로 표현되는 선형변환의 스케일 변환을 나타내는 값으로 해석할 수 있다.

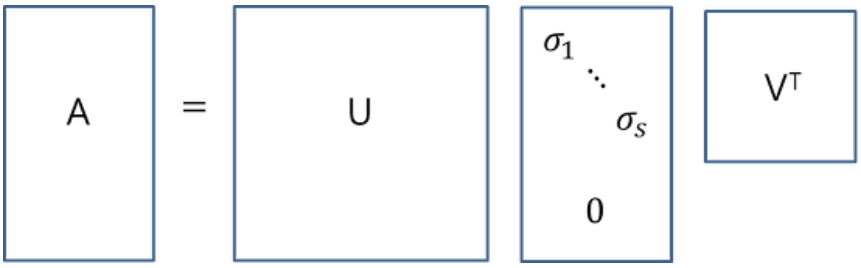
고유값분해(eigendecomposition)에서 나오는 고유값(eigenvalue)과 비교해 보면 고유값은 변환에 의해 불변인 방향벡터(-> 고유벡터)에 대한 스케일 factor이고, 특이값은 변환 자체의 스케일 factor로 볼 수 있다.

☞ 이 주제와 관련하여 조금 더 상상의 나래를 펴 보면, m x n 행렬 A는 n차원 공간에서 m 차원 공간으로의 선형변환이다. n차원 공간에 있는 원, 구 등과 같이 원형으로 된 도형을 A에 의해 변환시키면 먼저 V^T에 의해서는 회전만 일어나므로 도형의 형태는 변하지 않는다. 그런데 Σ에 의해서는 특이값의 크기에 따라서 원이 타원이 되거나 구가 럭비공이 되는 것과 같은 식의 형태변환이 일어난다 (n이 2차원인 원의 경우 첫번째 특이값 σ1은 변환된 타원의 주축의 길이, 두번째 특이값 σ2는 단축의 길이에 대응된다). 이후 U에 의한 변환도 회전변환이므로 도형의 형태에는 영향을 미치지 못한다.

만일 m>n이라면 0을 덧붙여서 차원을 확장한 후에 U로 회전을 시키는 것이고 m<n이라면 일부 차원을 없애버리고(일종의 투영) 회전을 시키는 셈이다. 결국 선형변환 A에 의한 도형의 변환결과는 형태적으로 보면 오로지 A의 특이값(singular value)들에 의해서만 결정된다는 것을 알 수 있다.

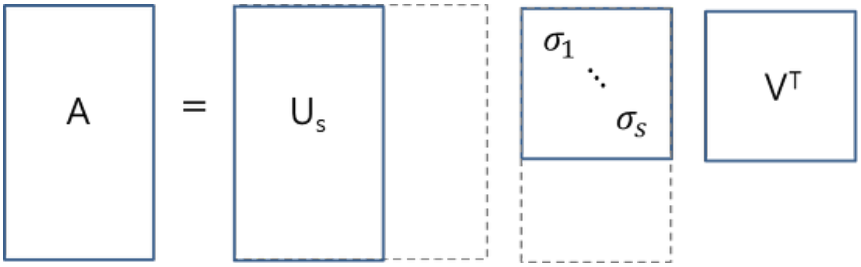
3. Reduced SVD와 행렬근사, 데이터 압축

아래와 같이 행렬 m×n 행렬 A를 SVD로 분해하는 것을 full SVD라 부른다 (단, m>n)

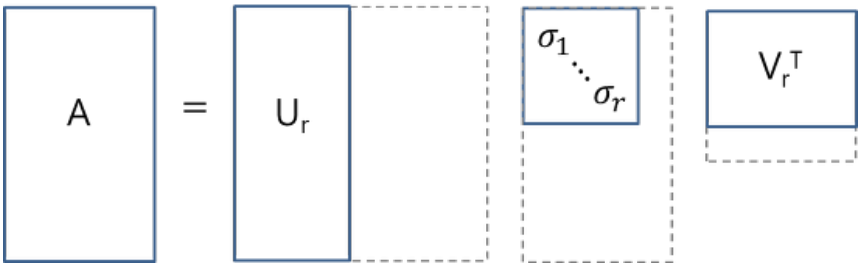


<그림 2> full SVD

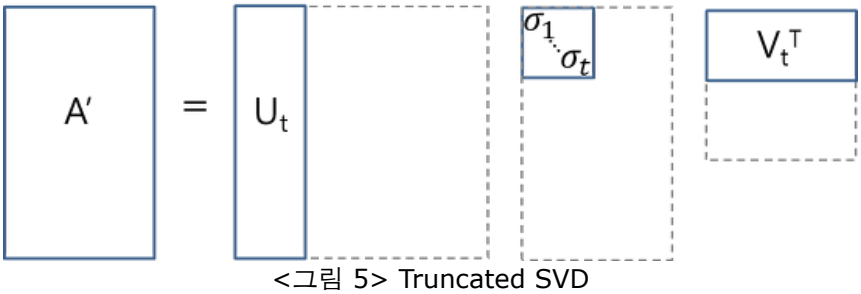
그런데, 실제로 이와같이 full SVD를 하는 경우는 드물며 아래 그림들과 같이 reduced SVD를 하는게 일반적이다 (s=n개의 singular value들 중 0이 아닌 것들의 개수가 r개, t<r라고 가정).



<그림 3> thin SVD



<그림 4> Compact SVD



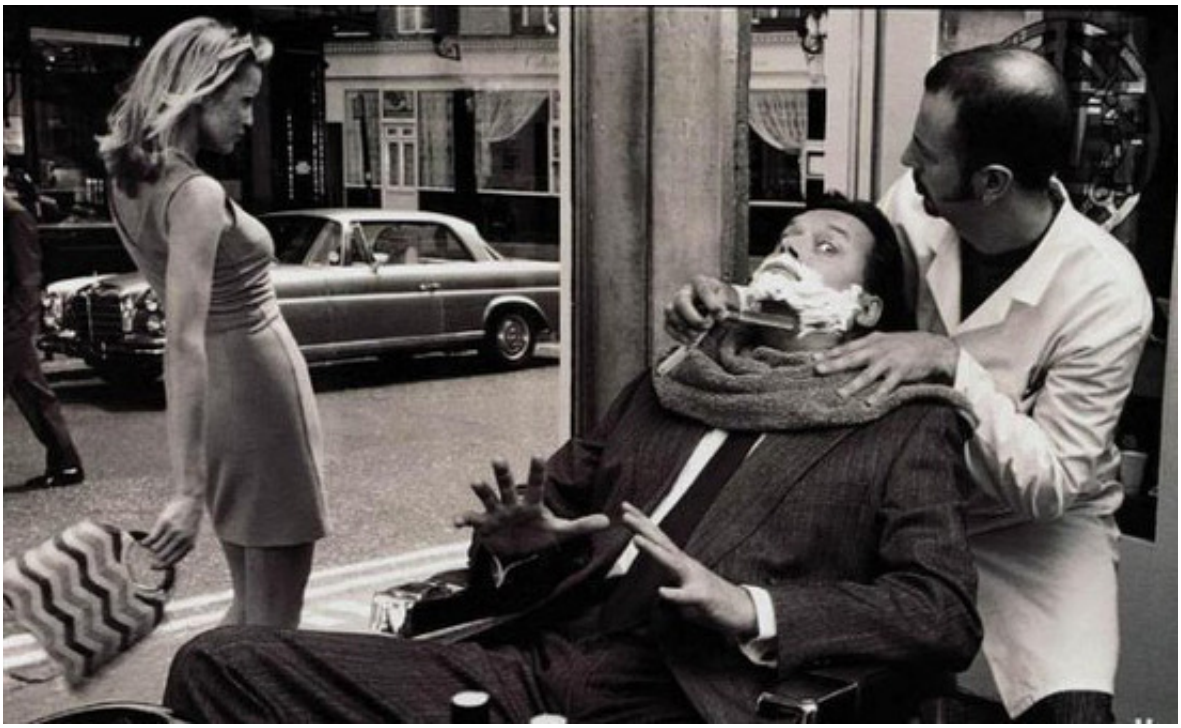
<그림 5> Truncated SVD

<그림 3>은 Σ에서 대각파트가 아닌 0으로 구성된 부분을 없애고 U에서는 이에 대응되는 열벡터들를 제거한 형태이고 (thin SVD라 부른다) <그림 4>는 비대각 원소들뿐만 아니라 0인 singular value들까지 모두 제거한 형태이다 (compact SVD). 이 때, 이렇게 계산된 A가 원래의 A와 동일한 행렬이 나옴은 쉽게 확인할 수 있다. 그러나, <그림 5>

의 경우는 0이 아닌 singular value까지 제거한 형태로서(truncated SVD) 이 경우에는 원래의 A가 보존되지 않고 A에 대한 근사행렬 A'이 나온다.

그런데, 이렇게 truncated SVD로 근사한 행렬 A'은 matrix norm ∥A-A'∥을 최소화시키는 rank t 행렬로서 데이터압축, 노이즈제거 등에 활용될 수 있다.

데이터 압축의 한 예로 아래와 같은 600 × 367 이미지를 SVD로 압축해 보자.



<그림 6> Wallis의 "dressed to kill" 광고 이미지들중 하나

이 이미지의 픽셀값을 원소값으로 하는 600 × 367 행렬 A를 잡고 SVD를 한 후, truncated SVD를 이용하여 근사행렬 A'을 구한 후 이를 다시 이미지로 표시해 보면 다음과 같다 (통상적인 m>n 형태로 만들기 위해 원래 이미지를 90도 회전시킨 후 SVD를 적용).

<그림 7> 100개의 singular value로 근사 (t = 100)

<그림 8> 50개의 singular value로 근사 (t = 50)

<그림 9> 20개의 singular value로 근사 (t = 20)

위 그림들 중 t=20인 경우의 데이터 압축율은 8.8%이다. 원래 이미지를 표현하기 위해서는 367*600 = 220,200의 메모리가 필요하다. 그런데, <그림 9> t = 20인 경우에는 600*20(U) + 20(Σ) + 20*367(V) = 19,360이므로 압축률은 19,360/220,200*100 = 8.8% 이다. 화질을 보면 좋은 압축 방법은 아님을 알 수 있지만 truncated SVD를 통한 데이터 근사가 원래의 데이터 핵심을 잘 잡아내고 있음을 알 수 있다.

4. 특이값 분해(SVD)와 pseudo inverse, 최소자승법(least square method)

pseudo inverse, 수도우 인버~스 (한글로 발음을 적고 보니 참 낯서네요).. 어쨌든 굳이 번역하자면 의사역행렬 정도이다.

선형시스템 Ax = b가 있을 때, 만일 A의 역행렬이 존재한다면 이 시스템의 해는 $x = A^{-1}b$ 로 손쉽게 구할 수 있다. 그러나 대부분의 실제 문제들에 있어서 역행렬이 존재하는 경우는 거의 없다고 봐도 무방하며 이런 경우 사용할 수 있는 것이 pseudo inverse이다. A의 pseudo inverse를 A^+라면 A의 역행렬이 없는 경우 Ax = b의 해는 $x = A^+b$와 같이 계산하고 이렇게 구한 x는 ∥Ax-b∥를 최소화하는 해가 된다. 이렇게 pseudo inverse를 이용하여 해를 구하는 것은 최소자승법, least square method와 같은 의미가 된다.

원래 역행렬(inverse matrix)은 정방행렬(square matrix)에 대해서만 정의된다. 하지만 pseudo inverse는 임의의 m x n 행렬에 대해서 정의될 수 있으며 특이값분해(SVD)는 pseudo inverse를 계산하는 가장 강력한(안정적인) 방법 중 하나이다.

행렬 A의 특이값분해(SVD)가 $A = U\Sigma V^T$라면 A의 pseudo inverse는 $V\Sigma^+U^T$로 계산되고 A^+로 표기한다 (단, Σ^+는 원래의 Σ에서 0이 아닌 singular value들의 역수를 취한 후 transpose를 시킨 행렬).

$$A=U\Sigma V^T$$

$$A^+=V\Sigma^+U^T$$

$$A=U\begin{pmatrix}\sigma_1 & & \\ & \ddots & \\ & & \sigma_s \\ & & 0\end{pmatrix}V^T \longrightarrow A^+=V\begin{pmatrix}1/\sigma_1 & & & \\ & \ddots & & \\ & & 1/\sigma_s & 0\end{pmatrix}U^T \quad \text{--- (4)}$$

U, V의 순서가 바뀌고 Σ도 m x n에서 n x m 행렬로 바뀜에 주의하자. 그리고 만일 특이값(singular value)들 중 0이 포함된 경우에는 0이 아닌 특이값들만 역수를 취하고 원래의 0은 Σ^+에도 그대로 0으로 놔둔다 (즉, 0이 아닌 경우만 역수).

[참고사항1]
만일 모든 singular value들이 양수이고 m≥n이면 A^+A는 n x n 단위행렬(identity matrix)이 된다. 만일 m≤n인 경우에는 AA^+가 단위행렬(m x m)이 된다. 그리고 만일 0인 singular value가 포함되면 어떤 순서로 곱해도 단위행렬은 나오지 않는다.

$$A^+A=(V\Sigma^+U^T)(U\Sigma V^T)=V\Sigma^+\Sigma V^T=VV^T=E_n \quad (m\geq n) \quad \text{--- (5)}$$

$$AA^+=(U\Sigma V^T)(V\Sigma^+U^T)=U\Sigma\Sigma^+U^T=UU^T=E_m \quad (m\leq n) \quad \text{--- (6)}$$

보통 우리가 선형연립방정식에서 pseudo inverse를 적용하는 경우는 m≥n인 즉, 식(데이터)의 개수가 미지수의 개수보다 많은 경우로서 Ax = b의 양변의 앞쪽에 A^+를 곱하여 $A^+Ax = A^+b \Rightarrow x = A^+b$의 형태로 x를 구하는 것이다.

m × n 행렬 A에 대해 SVD를 할 경우에는 보통 m>n인 행렬을 대상으로 하는 것이 일반적이다. m<n인 경우는 선형연립방정식으로 보면 미지수의 개수가 식의 개수가 많은 경우로서 해가 유일하게 결정되지 않는다. 이는 마치 점 2개를 가지고 평면을 결정하라는 문제와 같다. 따라서 특이값분해(SVD)를 통해 pseudo inverse를 구하는 것은 m>n인 행렬을 대상으로 한다고 생각하는게 좋다. m<n인 경우에도 pseudo inverse를 구할 수는 있지만 식 (6)과 같이 A의 뒤에 곱하는 형태로 사용해야 하므로 Ax=b 꼴 문제에는 유효하지 않다.

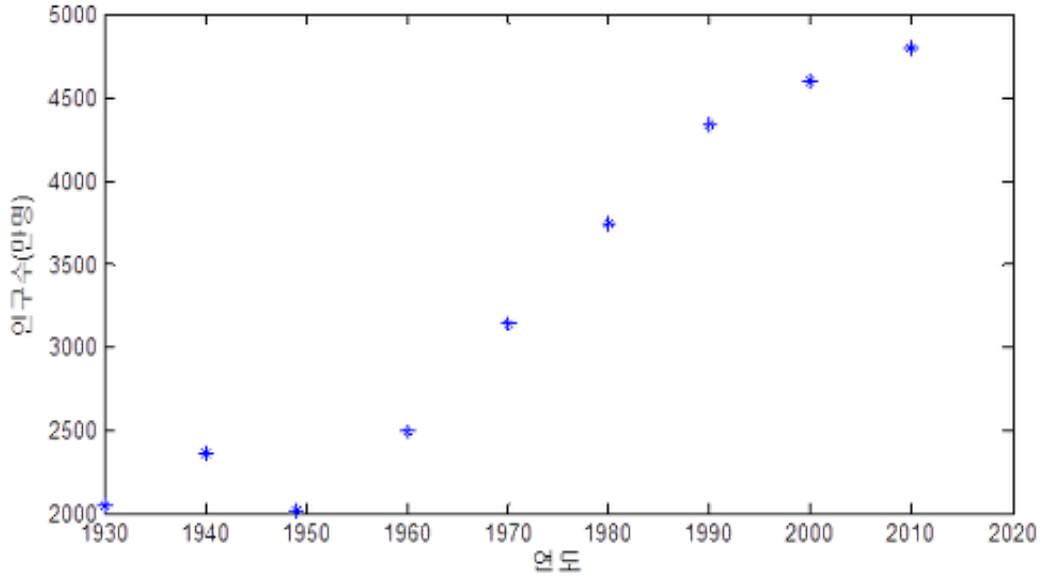
[참고사항2]
특이값이 0이 아니더라도 0에 매우 가까운 경우에는 노이즈로 치고 0으로 바꾼 후에 pseudo inverse를 구하는 것이 일반적이다. 즉, 어떤 행렬 A의 pseudo inverse를 SVD로 구하기 위해서는 먼저 특이값들을 구한 후에, 0에 매우 가까운 특이값들은 0으로 바꾸고 임계치 이상의 특이값들만 역수를 취해서 pseudo inverse를 구한다. 이 임계치를 SVD의 tolerance라고 부르는데, matlab에서 사용하는 tolerance 기본값은 1e-10 이다. 이는 앞서 설명한 truncated SVD와도 밀접한 관련이 있으며 어느 정도의 값을 노이즈로 볼 것인가에 따라서 즉, tolerance 값을 어떻게 줄 것인가에 따라서 결과 pseudo inverse 및 선형시스템의 해가 달라질 수 있다. 아래 예를 통해 tolerance 값에 따라서 선형시스템의 해가 어떻게 달라질 수 있는지 살펴보도록 하자.

SVD를 이용한 선형시스템 풀이 예제

예를 들어 우리나라의 1930년 ~ 2010년 사이의 10년 단위의 연도별 인구추이를 보고 2020년도 총인구수를 예측하는 문제를 생각해 보자 (자료출처: KOSIS 국가통계포털). 참고로 2012년 기준 현재 우리나라 총 인구수는 5,095만명이다.

(단위: 만명)

연도	1930	1940	1949	1960	1970	1980	1990	2000	2010	2020
인구	2,044	2,355	2,017	2,499	3,144	3,741	4,339	4,599	4,799	?



<그림 10> 우리나라 연도별 인구추이 그래프

우리나라 인구수 추이를 보면 계속 인구가 증가하다가 2000년 경부터 증가추세가 둔화되는 경향을 보이고 있다. 따라서, 우리나라 인구수 추이 모델을 2차 포물선으로 잡고 2020년도 국내 인구를 추정해 보자.

연도를 x = 1930, 1940, ..., 2010, 해당연도의 인구수를 y, 모델을 $y = ax^2 + bx + c$로 잡자.

$$\begin{aligned} ax_1^2+bx_1+c&=y_1 \\ ax_2^2+bx_2+c&=y_2 \\ &\vdots \\ ax_9^2+bx_9+c&=y_9 \end{aligned} \quad \text{--- (7)}$$

$$\begin{bmatrix} 1{,}930^2 & 1{,}930 & 1 \\ 1{,}940^2 & 1{,}940 & 1 \\ \vdots & \vdots & \vdots \\ 2{,}010^2 & 2{,}010 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 2{,}044 \\ 2{,}355 \\ \vdots \\ 4{,}799 \end{bmatrix}$$

$$A = \begin{bmatrix} 1{,}930^2 & 1{,}930 & 1 \\ 1{,}940^2 & 1{,}940 & 1 \\ \vdots & \vdots & \vdots \\ 2{,}010^2 & 2{,}010 & 1 \end{bmatrix},\ X = \begin{bmatrix} a \\ b \\ c \end{bmatrix},\ B = \begin{bmatrix} 2{,}044 \\ 2{,}355 \\ \vdots \\ 4{,}799 \end{bmatrix} \quad \text{--- (8)}$$

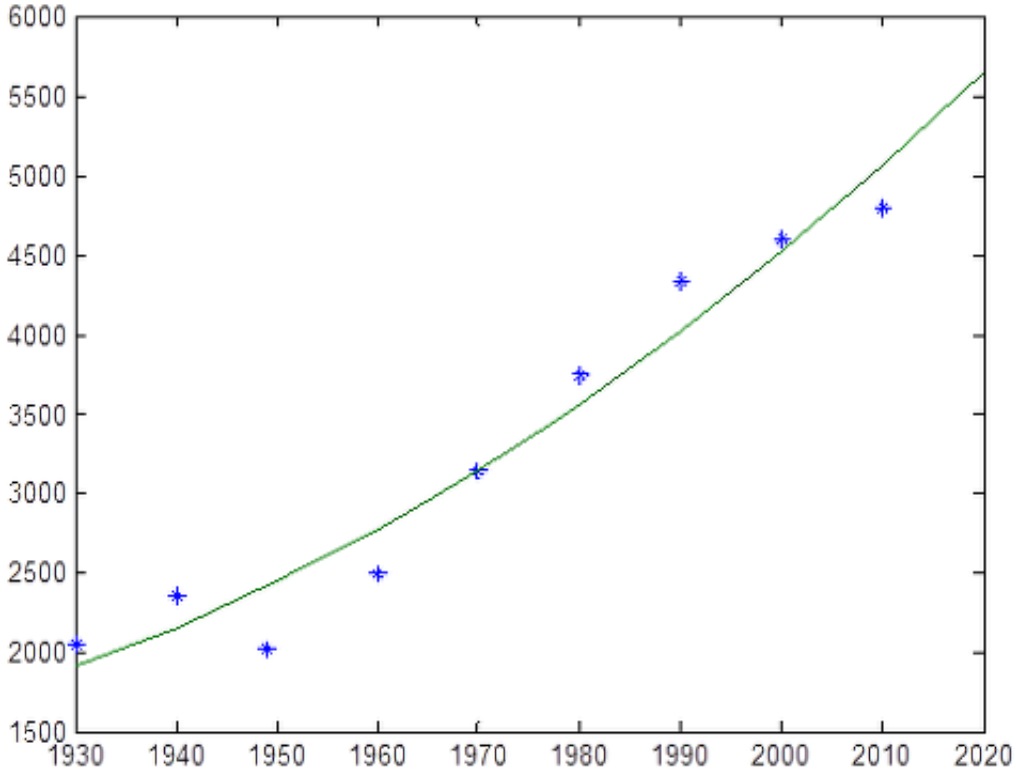
위와 같이 선형 연립방정식을 세우고 이를 행렬식 AX = B 형태로 표현했을 때, A의 역행렬이 존재한다면 a, b, c를 바로 결정할 수 있을 것이다. 하지만 위 <그림 10> 그래프의 모든 점을 지나는 포물선은 존재하지 않을 것이므로 선형연립방정식 (7)의 해는 당연히 존재하지 않고 A의 역행렬 또한 존재하지 않는다.

이런 경우 A의 pseudo inverse A^+를 구하여 $X = A^+B$로 X를 구하면 인구추이 데이터를 least square(최소자승)로 근사하는 포물선을 구할 수 있다.

실제로 행렬 A에 대해 특이값분해(SVD)를 해 보면 다음과 같은 3개의 singular value들을 얻을 수 있다 (A가 9 × 3 행렬이므로 singular value의 개수는 총 3개 = min(3, 9)).

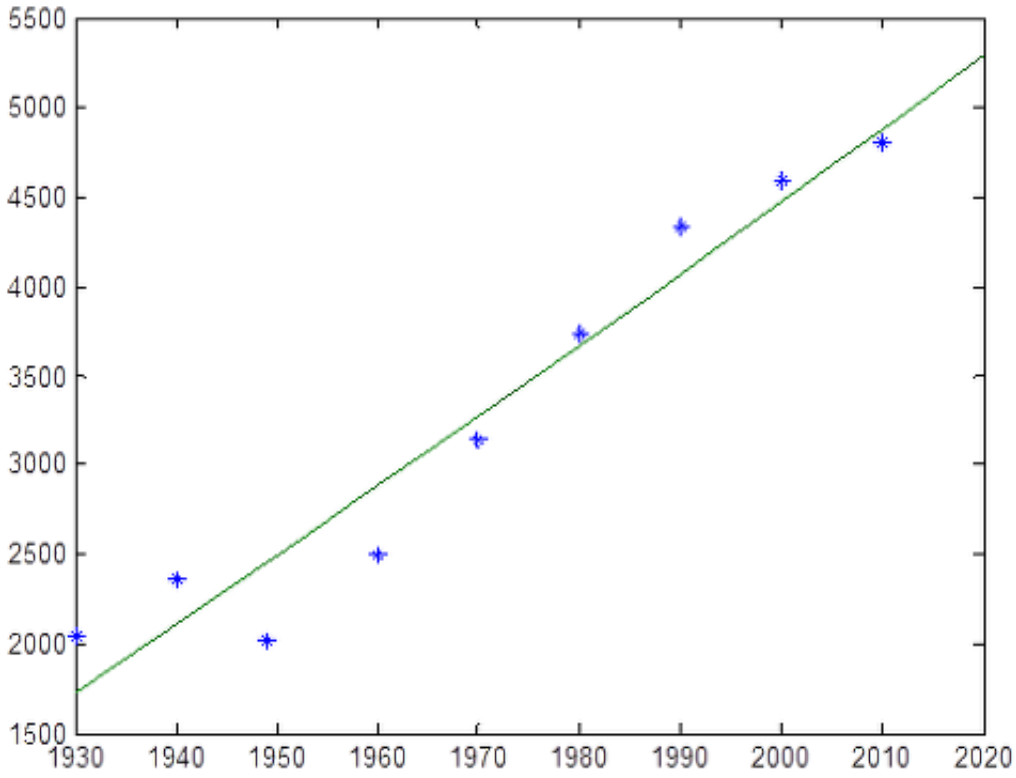
$$\sigma_1=11647428$$
$$\sigma_2=77.6794$$
$$\sigma_3=0.00045 \quad \text{--- (9)}$$

i) σ1, σ2, σ3를 모두 사용한 full SVD 경우
full SVD를 이용하여 pseudo inverse를 구했을 경우 계산된 포물선 그래프는 아래 그림과 같다. 구해진 포물선을 이용하여 2020년도 인구를 추정해 보면 약 5,645만명 정도가 나온다 (추정된 포물선 방정식: y = 0.21*x^2 - 802.59*x + 754923)



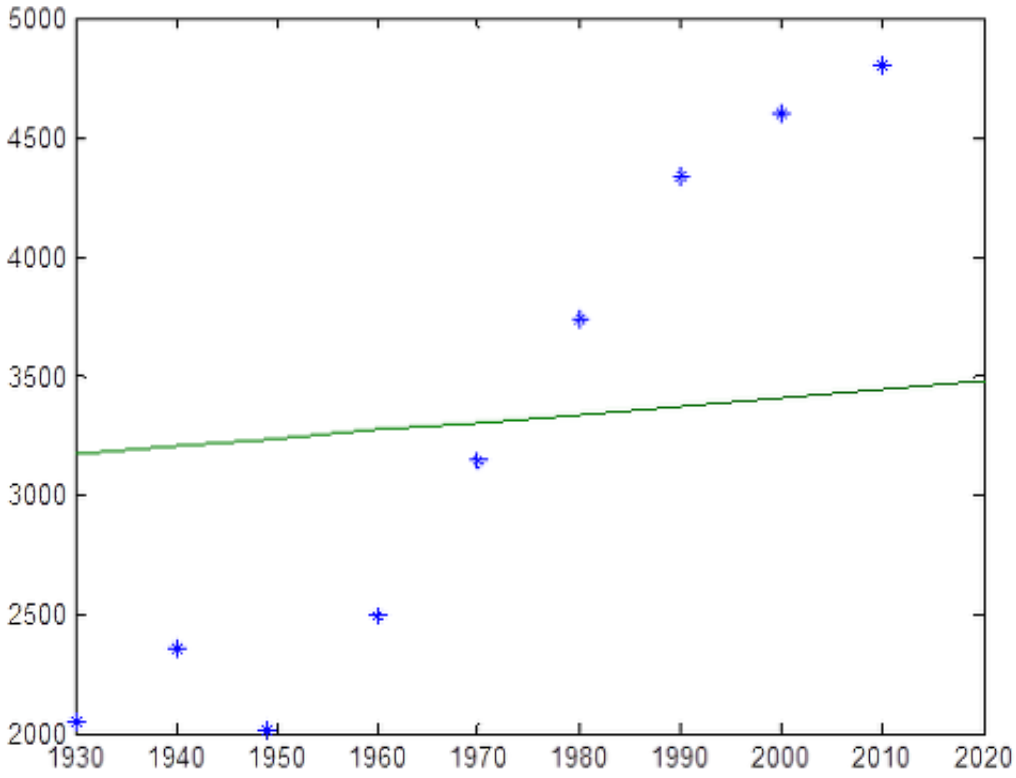
<그림 11> full SVD 근사

ii) σ1, σ2 만을 이용한 truncated SVD 경우
σ3=0으로 놓고, σ1, σ2 만을 이용해 pseudo inverse를 구했을 경우는 아래와 같은 포물선을 얻는다. full SVD에 비해 좀더 단순화된 모델로 데이터를 근사함을 볼 수 있다. 이 때, 이 포물선을 이용한 2020년도 추정 인구수는 5290만명이다 (추정된 포물선 방정식: y = 0.02*x^2 - 36.07*x - 0.04).



<그림 12> truncated SVD 근사 (t=2)

iii) σ1 만을 이용한 truncated SVD 경우
σ2=σ3=0으로 모두 날려버리고 σ1만을 이용해 pseudo inverse를 구해 포물선을 구한 결과는 아래 그림과 같다. ii) 경우보다도 훨씬 더 단순화된 모델링을 하는 것을 볼 수 있다 (너무 단순화가 심해서 데이터의 미세한 변화 특성을 대부분 잃어버림). 이 경우 예측된 2020년도 인구수는 3476만명이다 (추정된 포물선 방정식: $y = 0.00085*x^2$).



<그림 13> truncated SVD 근사 (t=1)

위 SVD 선형근사 예제에 대한 matlab 코드는 다음과 같다.

```
%% SVD population estimation

yr = [1930, 1940, 1949, 1960, 1970, 1980, 1990, 2000, 2010]';
pop = [2044, 2355, 2017, 2499, 3144, 3741, 4339, 4599, 4799]';

A = [yr.^2 yr ones(length(yr),1)];
B = pop;

[U D V] = svd(A);

% pseudo inverse with full SVD
D_inv = D;
D_inv(1,1) = 1/D(1,1);
D_inv(2,2) = 1/D(2,2);
D_inv(3,3) = 1/D(3,3);
A_pinv = V*D_inv'*U';
X = A_pinv*B;
pop_approx = A*X;
plot(yr, pop, '*', yr, pop_approx);

% pseudo inverse with truncated SVD (t=2)
D_inv = D;
D_inv(1,1) = 1/D(1,1);
D_inv(2,2) = 1/D(2,2);
```

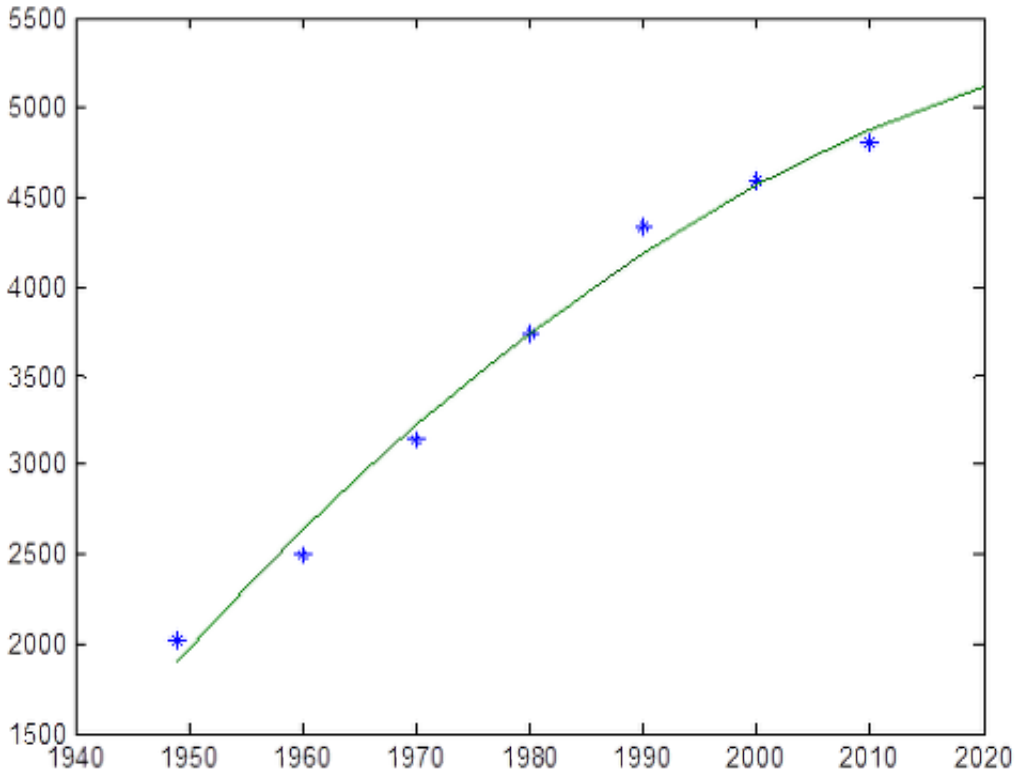
```
D_inv(3,3) = 0;
A_pinv = V*D_inv'*U';
X = A_pinv*B;
pop_approx = A*X;
figure; plot(yr, pop, '*', yr, pop_approx);

% pseudo inverse with truncated SVD (t=1)
D_inv = D;
D_inv(1,1) = 1/D(1,1);
D_inv(2,2) = 0;
D_inv(3,3) = 0;
A_pinv = V*D_inv'*U';
X = A_pinv*B;
pop_approx = A*X;
figure; plot(yr, pop, '*', yr, pop_approx);

return
```

그런데, 위 <그림 11>의 full SVD로 근사한 결과 그래프를 보면 2000년 경부터 우리나라 인구증가 속도가 조금씩 감소하고 있는 실제 경향이 제대로 반영되지 못함을 알 수 있다. 그 이유는 1950년대 6.25 전쟁으로 인해 인구수가 급격히 감소한 부분이 전체 포물선 근사에 영향을 미쳤기 때문이다. 즉, pseudo inverse를 이용해 데이터를 근사하는 것은 전체 데이터의 근사오차를 최소화시키는 최소자승법(least square method) 근사이기 때문에 이런 문제점이 발생한다 (이와 같이 outlier가 포함된 데이터는 최소자승법으로는 잘 근사할 수 없다).

만일 1930, 1940년도 데이터는 버리고 1950년도 데이터부터만 사용하여 다시 인구수 변화를 근사하면 아래 그림과 같은 결과가 나온다. 훨씬 그럴듯한 근사결과가 나옴을 확인할 수 있으며 이 근사 결과를 이용하여 2020년도 우리나라 인구수를 예측해 보면 약 5,116만명이 나온다.



<그림 14> outlier를 제거한 full SVD 근사

그런데 정말 2020년도 우리나라 인구수가 5,116만명이 될까요?? ^^

[선형대수학 #1] 주요용어 및 기본공식
[선형대수학 #2] 역행렬과 행렬식(determinant)
[선형대수학 #3] 고유값과 고유벡터 (eigenvalue & eigenvector)
[선형대수학 #4] 특이값 분해(Singular Value Decomposition, SVD)의 활용
[선형대수학 #5] 선형연립방정식 풀이
[선형대수학 #6] 주성분분석(PCA)의 이해와 활용

by 다크 프로그래머

'수학 이야기' 카테고리의 다른 글

[선형대수학 #5] 선형연립방정식 풀이 (31) 2013.10.29
[선형대수학 #4] 특이값 분해(Singular Value Decomposition, SVD)의 활용 (136)