SVD Method.

임의의 m×n 크기 행렬은 아래와 같이 분해될 수 있다. (SVD theorem)

$\underset{m\times n}{A} = \underset{m\times m}{U}\ \underset{m\times n}{\Sigma}\ \underset{n\times n}{V^T}$ s.t. $U = \begin{pmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_m \\ | & | & & | \end{pmatrix}$ $V = \begin{pmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_n \\ | & | & & | \end{pmatrix}$

u_i: m×1 vector, $\|u_i\|=1$, $u_i' u_j = 0\ \forall i\neq j$

v_i: n×1 vector, $\|v_i\|=1$, $v_i' v_j = 0\ \forall i\neq j$

$\Sigma = \begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_m \end{pmatrix}$ $\Sigma_{ii} = \lambda_i > 0$, $\Sigma_{ij} = 0\ \forall i\neq j$

그렇다면 A를 아는 것 $\Leftrightarrow$ U, Σ, V^T를 아는 것. —— ①

또한, $A = U\Sigma V^T = \begin{pmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_m \\ | & | & & | \end{pmatrix}\begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_m \end{pmatrix}\begin{pmatrix} - v_1' - \\ - v_2' - \\ \vdots \\ - v_n' - \end{pmatrix} = \sum \lambda_i u_i v_i'$ —— ㄱ

이제, λ_i를 내림차순으로 정렬한 Σ_{NEW}와, ~~해당 λi에 대응하는~~ U와 V의 column vector들을 대응하는 λ_i의 순서에 맞게 정렬한 U_{NEW}, V_{NEW}를 구성하자.

(가령, U,Σ,V: 3×3 크기일 때, $\lambda_3 > \lambda_2 > \lambda_1$ 이면 $\Sigma_{NEW} = \begin{pmatrix} \lambda_3 & & 0 \\ & \lambda_2 & \\ 0 & & \lambda_1 \end{pmatrix}$, $U_{NEW} = \begin{pmatrix} | & | & | \\ u_3 & u_2 & u_1 \\ | & | & | \end{pmatrix}$, $V_{NEW} = \begin{pmatrix} | & | & | \\ v_3 & v_2 & v_1 \\ | & | & | \end{pmatrix}$)

그리고 $k<m$, $k<n$ 을 만족하는 어떤 자연수 k가 있으면,

U_{NEW}, V_{NEW}는 앞에서부터 k개의 column vector만 남기고 다 자른 $\hat{U}$, $\hat{V}$를,

Σ_{NEW}는 좌상단부터 k×k 크기 행렬만 남긴 $\hat{\Sigma}$를 만들자.

그러면, $\hat{U}\hat{\Sigma}\hat{V}^T = \sum \hat{\lambda}_i \hat{u}_i \hat{v}_i' \approx \underbrace{\sum \lambda_j u_j v_j = U\Sigma V^T = A}_{\text{by ㄱ}}$. —— ②

($\hat{\lambda}_i$: i번째로 큰 λ, $\hat{u}_i$, $\hat{v}_i$는 $\hat{\lambda}_i$에 대응되는 u, v)

($\because$ $\|u\|=\|v\|=1$ 로 u와 v 둘의 크기는 일정하다. 그러므로 $u_i v_i'$ 로 표현되는 matrix의 크기도 고만고만함.

결국, $A = \sum \lambda_i u_i v_i'$ 에서는 가중치인 λ_i가 클수록 $u_i v_i'$가 A를 구성하는 지분이 커짐.

근데 $\hat{\lambda}_i$는 가장 큰 k개의 λ에 대응되는 것이니까,

$\hat{\lambda}_i \hat{u}_i \hat{v}_i'$는 기존의 A를 구성하던 $\lambda u v'$ 중 지분이 가장 컸던 k개에 해당.)

∴ ①, ②에 의해, $\hat{U}$, $\hat{\Sigma}$, $\hat{V}^T$를 알면 A를 (근사) 추정할 수 있다. — ③.

그러나, utility matrix는 sparse하다. (일부 원소에 대해서만 data가 있다.)
(가령, $\begin{pmatrix} ? & ? & 2 \\ 1 & ? & ? \\ ? & 3 & ? \end{pmatrix}$ 이라고 봐.)

따라서, 이 matrix로는 U, Σ, V^T를 구할 수 없고, $\hat{U}$, $\hat{\Sigma}$, $\hat{V}^T$도 구할 수 없다.

$\hat{U}$, $\hat{\Sigma}$, $\hat{V}^T$를 "추정"해야 한다!

추정법. $\hat{U} = \begin{pmatrix} u_{11} & u_{12} & & \\ & & \ddots & \\ & & & u_{mk} \end{pmatrix}$, $\hat{V} = \begin{pmatrix} v_{11} & v_{12} & & \\ & & \ddots & \\ & & & v_{nk} \end{pmatrix}$, $\hat{\Sigma} = \begin{pmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ 0 & & \ddots \ \lambda_k \end{pmatrix}$ 일 때.

~~우리는 A의 몇몇 원소들 (a_{ij} 몇 개)을 알고~~

$\hat{U}\hat{\Sigma}\hat{V}^T$의 (i,j)번째 원소는 $\sum_{x=1}^{k} \lambda_x u_{ix} v_{jx}$ 로 계산된다.

그렇다면, $\sum \lambda_x u_{ix} v_{jx}$가 우리가 알고 있는 A의 원소들과 가까우면 된다.
그러면 $\hat{U}$, $\hat{\Sigma}$, $\hat{V}^T$가 잘 추정된 것이다.

$$\therefore \ \text{minimize} \sum_{i,j} \underbrace{\left(a_{ij} - \sum_{x=1}^{k} \lambda_x u_{ix} v_{jx}\right)^2}_{\text{실제 value와 } \hat{U}, \hat{\Sigma}, \hat{V}^T \text{로 예측한 것 간의 차이}}$$

s.t $a_{ij} \in A$를 알고 있음.

이것을 최소화하는 λ_x, u_{ix}, v_{jx} 값들을 갖는 행렬을 $\hat{U}$, $\hat{\Sigma}$, $\hat{V}^T$의 추정치로 보자! — ④

③, ④에 의해, 추정된 $\hat{U}$, $\hat{\Sigma}$, $\hat{V}^T$로부터 A를 추정할 수 있다.