Recommender System using Matrix Factorization

Jia-Wei Liao*, Kuok-Tong Ng, Yi-Cheng Hung

Department of Applied Mathematics
National Yang Ming Chiao Tung University

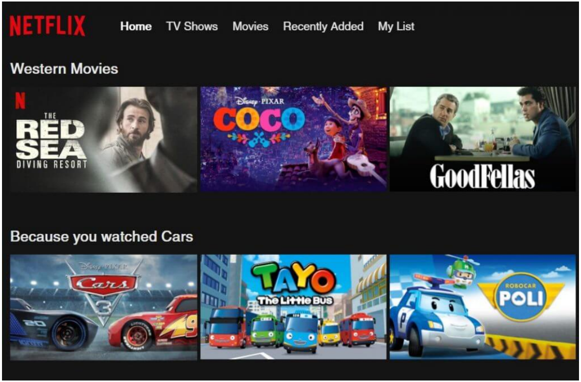
January 26, 2023

Outline

Introduction to Recommender System

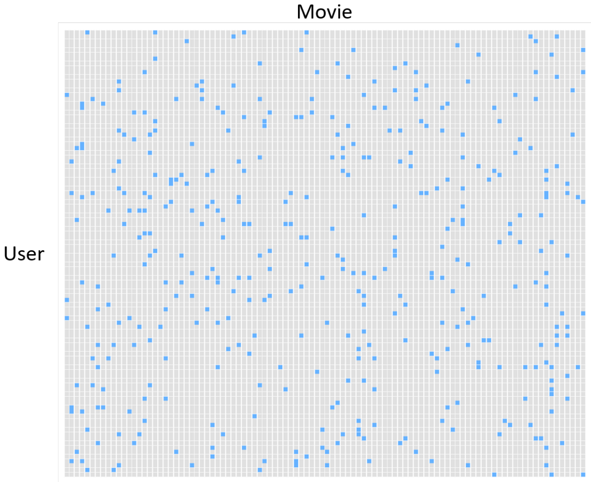
A competition held by Netflix in 2006.

- 100,480,570 ratings that 480,189 users gave to 17,770 movies.
- 10% improvement then gain 1 million dollar prize.



Introduction to Recommender System

(user, movie)	rating
(1, 5)	4
(1, 18)	1
(1, 32)	3
(1, 44)	2
(2, 22)	5
(2, 90)	2
(3, 49)	3
(3, 56)	4
(3, 70)	5
(3, 94)	1
⋮	⋮



Idea of Low-rank Approximation



Action	0.2	1.2	1.8	1.8
Romance	1.68	0.5	0.1	0.3
Science Fiction	0.2	1.8	0.6	1.6
Fantasy	0.48	0.8	0.4	0.4

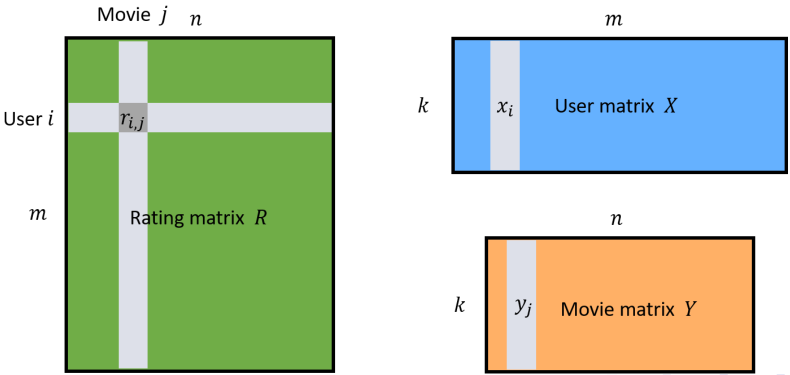
	A.	R.	S.	F.
Sam	0.8	1.6	1.8	1.2
Jack	1.1	0.3	1.2	1.6
Andy	1.3	0.5	1.6	0.8

5.07

Idea of Low-rank Approximation

Given $R \in \mathbb{R}^{m \times n}$ is rating matrix, where m is number of user, and n is number of movie. Our goal is going to find the feature vector of user x_i and feature vector of movie y_j such that

$$r_{i,j} \approx x_i^\top y_j, \text{ for all } i, j$$



Problem

Given $R \in \mathbb{R}^{m\times n}$ is rating matrix, Our goal is going to find the user matrix $X \in \mathbb{R}^{k\times m}$ and feature matrix $Y \in \mathbb{R}^{k\times n}$ such that

$$R \approx X^\top Y.$$

Question

1. How to find X and Y ?
2. How to approximate R ?
3. How to compute with big data?

Outline

Singular Value Decomposition (SVD)

SVD

Let $A \in \mathbb{R}^{m\times n}$. Then there exist orthogonal matrices $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ such that $U^\top AV = \Sigma$ is a diagonal matrix, where

$$\Sigma_{ij} = \begin{cases} \sigma_i, & i = j \\ 0, & i \neq j \end{cases}, \text{ with } \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0,$$

and $r = \text{rank}(A)$.

Singular Value Decomposition (SVD)

Separates A into r pieces rank 1 by SVD

$$A = \begin{bmatrix} u_1 & \cdots & u_m \end{bmatrix} \left[\begin{array}{ccc|c} \sigma_1 & & & \\ & \ddots & & O_{r\times(n-r)} \\ & & \sigma_r & \\ \hline & O_{(m-r)\times r} & & O_{(m-r)\times(n-r)} \end{array}\right] \begin{bmatrix} v_1^\top \\ \vdots \\ v_n^\top \end{bmatrix}$$

$$= \sum_{i=1}^{r} \sigma_i u_i v_i^\top.$$

Rank k approximate

Let $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^\top$. If B has rank k, then

$$\|A - A_k\|_* \leq \|A - B\|_*.$$

Matrix Norm

Matrix norm

1. **Spectral norm:**

$$\|A\|_2 = \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \sigma_1,$$

2. **Frobenius norm:**

$$\|A\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} |a_{ij}|^2} = \sqrt{\sigma_1^2 + \cdots + \sigma_r^2},$$

3. **Nuclear norm:**

$$\|A\|_N = \sigma_1 + \cdots + \sigma_r.$$

Outline

Idea

Rank k approximation

Given $k \in \mathbb{N}$ with $1 \leq k < r$.

$$R = U\Sigma V^\top \approx U_k {D_k}^2 V_k = (D_k U_k^\top)^\top (D_k V_k^\top) = X^\top Y$$

where $\tilde{U} = [u_1, ..., u_k]$, $\tilde{V} = [v_1, ..., v_k]$ and $D = \text{diag}(\sqrt{\sigma_1}, ..., \sqrt{\sigma_k})$.

Question

Does it work on the largest dataset?

Matrix Factorization

First, We consider the minimization problem with regularization.

$$\min_{X,Y} \|R - X^\top Y\|_F^2 + \lambda \left(\|X\|_F^2 + \|Y\|_F^2\right),$$

where

- $R \in \mathbb{R}^{m\times n}$, $X \in \mathbb{R}^{k\times m}$ and $Y \in \mathbb{R}^{k\times n}$,
- $\lambda > 0$ is a parameter.

Remark

$$\begin{aligned}
&\min_{X\in\mathbb{R}^{k\times m}, Y\in\mathbb{R}^{k\times n}} \|R - X^\top Y\|_F^2 + \lambda \left(\|X\|_F^2 + \|Y\|_F^2\right) \\
&= \min_{Z:\operatorname{rank} Z\le k} \|R - Z\|_F^2 + 2\lambda\|Z\|_*.
\end{aligned}$$

Alternative Least Square (ALS)

Given initials X_0, Y_0, for $t = 0, 1, 2, \ldots$, we solve the following two sub-problem alternatingly:

$$X_{t+1} = \arg\min_{X} \|R - X^\top Y_t\|_F^2 + \lambda \|X\|_F^2,$$
$$Y_{t+1} = \arg\min_{Y} \|R - {X_{t+1}}^\top Y\|_F^2 + \lambda \|Y\|_F^2.$$

We iterate until convergence.

Solving Alternative Least Square

Define

$$F(X) = \|R^\top - {Y_t}^\top X\|_F^2 + \lambda \|X\|_F^2.$$

Then

$$\nabla F(X) = -2Y_t(R^\top - {Y_t}^\top X) + 2\lambda X.$$

Let $\nabla F(X) = 0$, we have

$$(Y_t {Y_t}^\top + \lambda I)X = Y_t R^\top.$$

Therefore, we obtain the solution

$$X_{t+1} = (Y_t {Y_t}^\top + \lambda I)^{-1} Y_t R^\top.$$

Solving Alternative Least Square

Similarly, we can get

$$Y_{t+1} = (X_{t+1}{X_{t+1}}^\top + \lambda I)^{-1} X_{t+1} R.$$

Therefore, the iterative scheme can be posed as follows:

1. $X_{t+1} = \left(Y_t {Y_t}^\top + \lambda I\right)^{-1} Y_t R^\top,$
2. $Y_{t+1} = \left(X_{t+1} {X_{t+1}}^\top + \lambda I\right)^{-1} X_{t+1} R.$

Matrix Factorization

Let $\Omega = \{(i,j) \mid R_{i,j} > 0\}$ and

$$[\mathcal{P}_\Omega(A)]_{i,j} = \begin{cases} A_{i,j} & \text{if } (i,j) \in \Omega \\ 0 & \text{if } (i,j) \notin \Omega \end{cases}.$$

Problem

$$\min_{X,Y} \|\mathcal{P}_\Omega(R - X^\top Y)\|_F^2 + \lambda \left(\|X\|_F^2 + \|Y\|_F^2\right),$$

where

- $R \in \mathbb{R}^{m \times n}$, $X \in \mathbb{R}^{k \times m}$, and $Y \in \mathbb{R}^{k \times n}$,
- $\lambda > 0$ is a parameter.

Matrix Factorization

Problem

$$\min_{X,Y} \|\mathcal{P}_\Omega(R - X^\top Y)\|_F^2 + \lambda \left(\|X\|_F^2 + \|Y\|_F^2\right).$$

Its equivalence to

$$\min_{x_i,y_j} \sum_{(i,j)\in\Omega} (r_{i,j} - x_i^\top y_j)^2 + \lambda \left(\sum_{i=1}^{m} \|x_i\|_2^2 + \sum_{i=1}^{n} \|y_j\|_2^2\right),$$

where $x_i, y_j \in \mathbb{R}^{k\times 1}$.

Solving Alternative Least Square

For fixed i, we define

$$F(x_i) = \sum_{(i,j)\in\Omega} (r_{i,j} - y_j^\top x_i)^2 + \sum_{i=1}^{m} \|x_i\|_2^2.$$

Then

$$\nabla F(x_i) = -2 \sum_{(i,j)\in\Omega} y_j (r_{i,j} - y_j^\top x_i) + 2\lambda x_i.$$

Letting $\nabla F(x_i) = 0$, we have

$$\left(\sum_{(i,j)\in\Omega} y_j y_j^\top + \lambda I \right) x_i = r_{i,j} y_j.$$

Solving Alternative Least Square

Algorithm (ALS)

1. Initial $x_1, x_2, ..., x_m, y_1, y_2, ..., y_n$.
2. For $i = 1, 2, ..., m$:

$$\left(\sum_{(i,j)\in\Omega} y_j y_j^\top + \lambda I \right) x_i = r_{i,j} y_j.$$

3. For $j = 1, 2, ..., n$:

$$\left(\sum_{(i,j)\in\Omega} x_i x_i^\top + \lambda I \right) y_j = r_{i,j} x_i.$$

4. Repeat 2, 3 until convergence.

Summary of ALS

1. The left-hand side of the linear system is sum of the rank 1 matrix.
2. There are $(m+n)$ linear systems with $k \times k$.
3. The time complexity of the direct method is $O(|\Omega|k^2 + (m+n)k^3)$.

Soft Impute Alternative Least Square (SIALS)

Problem

$$\min_{X,Y} \|\mathcal{P}_\Omega(R - X^\top Y)\|_F^2 + \lambda \left(\|X\|_F^2 + \|Y\|_F^2\right).$$

Notice that

$$\mathcal{P}_\Omega(R - X^\top Y) = \mathcal{P}_\Omega(R) - \mathcal{P}_\Omega(X^\top Y) + X^\top Y - X^\top Y.$$

Let

$$\tilde{R} = \mathcal{P}_\Omega(R) - \mathcal{P}_\Omega(X^\top Y) + X^\top Y.$$

Then the problem becomes

$$\min_{X,Y} \|\tilde{R} - X^\top Y\|_F^2 + \lambda \left(\|X\|_F^2 + \|Y\|_F^2\right).$$

Soft Impute Alternative Least Square (SIALS)

SIALS method

Given an initial X_0, Y_0, for $t = 0, 1, \ldots$, we solve the following two sub-problem alternatingly:

$$\begin{aligned}
\tilde{R}_{t+\frac{1}{2}} &= \mathcal{P}_\Omega(R) - \mathcal{P}_\Omega({X_t}^\top Y_t) + {X_t}^\top Y_t, \\
X_{t+1} &= \left(Y_t {Y_t}^\top + \lambda I\right)^{-1} Y_t \tilde{R}_{t+\frac{1}{2}}^\top, \\
\tilde{R}_{t+1} &= \mathcal{P}_\Omega(R) - \mathcal{P}_\Omega({X_{t+1}}^\top Y_t) + {X_{t+1}}^\top Y_t, \\
Y_{t+1} &= \left(X_{t+1} {X_{t+1}}^\top + \lambda I\right)^{-1} X_{t+1} \tilde{R}_{t+1}.
\end{aligned}$$

We iterate until convergence is achieved.

Solving SIALS by SVD

Idea (rank k approximation) [1]

$$R = U\Sigma V^\top \approx U_k D_k^2 V_k = (D_k U_k^\top)^\top (D_k V_k^\top) = X^\top Y.$$

Goal

Given $U_0 \in \mathbb{R}^{m\times k}$ with orthonormal columns, $D_0 = I_{k\times k}$, and $V_0 = O_{n\times k}$. We use the Iterative method to find the suitable $X = D_t {U_t}^\top$, $Y = D_t {V_t}^\top$ such that

$$\min_{X,Y} \|P_\Omega(R - X^\top Y)\|_F^2 + \lambda\left(\|X\|_F^2 + \|Y\|_F^2\right).$$

[1] T. Hastie, R. Mazumder, J.-D. Lee and R. Zadeh, Matrix Completion and Low-Rank SVD via Fast Alternating Least Squares, Journal of Machine Learning Research (2015).

Soft Impute ALS by SVD

Given $U_0 \in \mathbb{R}^{m \times k}$ with orthonormal columns, $D_0 = I_{k \times k}$, and $V_0 = O_{k \times k}$. Let $X_0 = D_0 {U_0}^\top$ and $Y_0 = D_0 {V_0}^\top$. For $t = 0, 1, \ldots$ do

1. $\tilde{R}_{t+\frac{1}{2}} = \mathcal{P}_\Omega(R) - \mathcal{P}_\Omega({X_t}^\top Y_t) + {X_t}^\top Y_t$
2. $X_{t+\frac{1}{2}} = \left(Y_t {Y_t}^\top + \lambda I\right)^{-1} Y_t \tilde{R}^\top_{t+\frac{1}{2}}$
3. Find the SVD of $D_t X_{t+\frac{1}{2}}$, then get $U_{t+1}, D_{t+\frac{1}{2}}$
4. $X_{t+1} = D_{t+\frac{1}{2}} {U_{t+1}}^\top$
5. $\tilde{R}_{t+1} = \mathcal{P}_\Omega(R) - \mathcal{P}_\Omega({X_{t+1}}^\top Y_t) + {X_{t+1}}^\top Y_t$
6. $Y_{t+\frac{1}{2}} = \left(X_{t+1} {X_{t+1}}^\top + \lambda I\right)^{-1} X_{t+1} \tilde{R}_{t+1}$
7. Find the SVD of $D_{t+\frac{1}{2}} Y_{t+\frac{1}{2}}$, then get V_{t+1}, D_{t+1}
8. $Y_{t+1} = D_{t+1} {V_{t+1}}^\top$

Soft Impute ALS

Let $S = \mathcal{P}_\Omega(R) - \mathcal{P}_\Omega({X_t}^\top Y_t)$. We have $\tilde{R} = S + {X_t}^\top Y_t$.
We use $X_t = D_t {U_t}^\top$ and $Y_t = D_t {V_t}^\top$ to plug in equation 2. Then

$$X_{t+\frac{1}{2}} = \left(D_t^2 + \lambda I\right)^{-1} D_t {V_t}^\top S^\top + \left(D_t^2 + \lambda I\right)^{-1} {D_t}^2 X_t.$$

Similarly, we also have

$$Y_{t+\frac{1}{2}} = \left(D_{t+\frac{1}{2}}^2 + \lambda I\right)^{-1} D_{t+\frac{1}{2}} {U_{t+1}}^\top S + \left(D_{t+\frac{1}{2}}^2 + \lambda I\right)^{-1} {D_{t+\frac{1}{2}}}^2 Y_t.$$

Solving SIALS by SVD

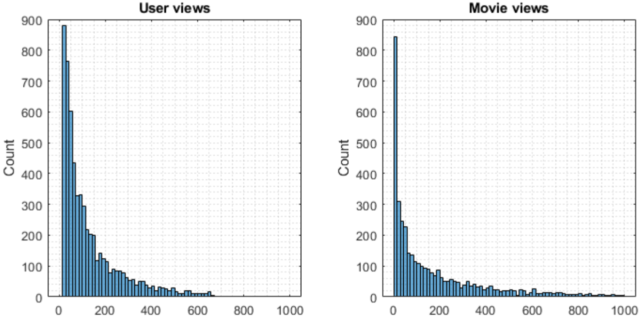
Algorithm (SIALS)

1. Initial: $R \in \mathbb{R}^{m\times n}, U \in \mathbb{R}^{m\times k}$ with orthonormal columns, $D = I_{k\times k}$, $V = O_{n\times k}$, $X = DU^\top$ and $Y = DV^\top$.
2. $S \leftarrow \mathcal{P}_\Omega(R) - \mathcal{P}_\Omega(X^\top Y)$
3. $\tilde{X} \leftarrow \left(D^2 + \lambda I\right)^{-1} DV^\top S^\top + \left(D^2 + \lambda I\right)^{-1} D^2 X$
4. $U, \tilde{D} \leftarrow \mathrm{SVD}(D\tilde{X})$, $X \leftarrow \tilde{D}U^\top$
5. $S \leftarrow \mathcal{P}_\Omega(R) - \mathcal{P}_\Omega(X^\top Y)$
6. $\tilde{Y} \leftarrow \left(D^2 + \lambda I\right)^{-1} DU^\top S + \left(D^2 + \lambda I\right)^{-1} D^2 Y$
7. $D, V \leftarrow \mathrm{SVD}(\tilde{D}\tilde{Y})$, $Y \leftarrow \tilde{D}V^\top$
8. Repeat 2-7 until convergence.
9. Output: U, V, D

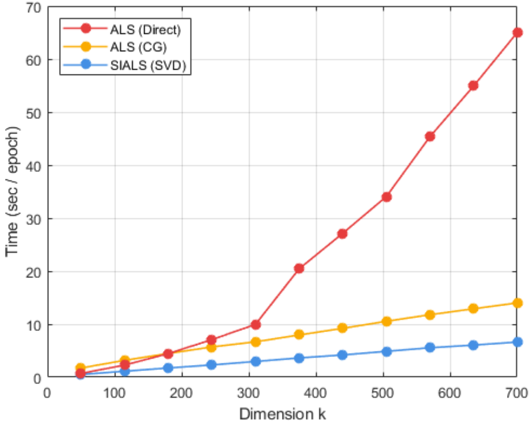
Outline

Information of Dataset: ml-1m

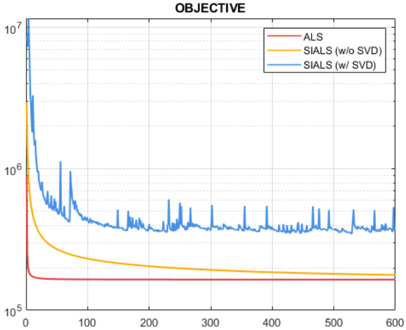
- number of data: 900,188
- number of user: 6,040
- number of movie: 3,952
- sparsity: 3.77%
- train-test split: 9 to 1



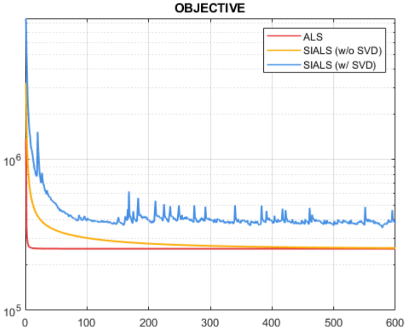
Time Elapsed



Objective



$\lambda = 5,\ d = 50$



$\lambda = 20,\ d = 50$

Metrics

We use the following metrics to evaluate our methods:

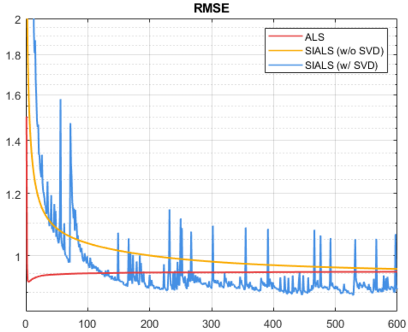
1. **Root Mean Square Error (RMSE):**

$$\mathrm{RMSE}_{\Omega}(X, Y) = \sqrt{\frac{\sum_{(i,j)\in\Omega} |X_{i,j} - Y_{i,j}|^2}{|\Omega|}}$$

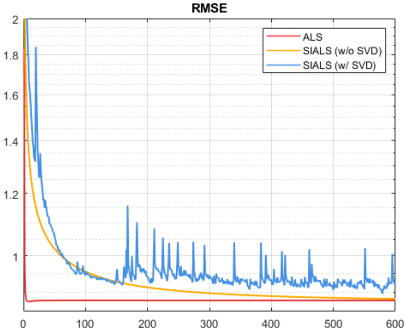
2. **Mean Absolute Error (MAE)**:

$$\mathrm{MAE}_{\Omega}(X, Y) = \frac{\sum_{(i,j)\in\Omega} |X_{i,j} - Y_{i,j}|}{|\Omega|}$$

Evaluation



$\lambda = 5, d = 50$



$\lambda = 20, d = 50$

Evaluation

Metrics	RMSE		MAE	
Type	Training	Testing	Training	Testing
ALS	0.5542	0.9545	0.4363	0.7385
SIALS (w/o SVD)	0.5745	0.9624	0.4517	0.7441
SIALS (w/ SVD)	0.8928	**0.9099**	0.7017	0.7143

$\lambda = 5$, $d = 50$ with 600 epochs

Metrics	RMSE		MAE	
Type	Training	Testing	Training	Testing
ALS	0.7288	**0.8778**	0.5727	0.6889
SIALS (w/o SVD)	0.7330	0.8820	0.5749	0.6913
SIALS (w/ SVD)	0.8980	0.9147	0.7101	0.7229

$\lambda = 20$, $d = 50$ with 600 epochs

ml-10m dataset

- number of data: 9,000,048
- number of user: 71,567
- number of movie: 65,133
- sparsity: 0.19%
- train-test split: 9 to 1

Metrics	RMSE		MAE	
Type	Training	Testing	Training	Testing
ALS	0.7287	**0.8453**	0.5453	0.6479
SIALS (w/ SVD)	0.9301	0.9454	0.7171	0.7286

$\lambda = 50$, $d = 100$ with 500 epochs

Prediction of SIALS

We compute $\hat{r}_{i,j} = x_i^\top y_j$, and list the prediction as the following:

(user, movie)	prediction	rounding	rating
(308, 1707)	1.5211	2	2
(990, 89)	2.7883	3	3
(2247, 2291)	4.0016	4	3
(2454, 595)	3.8591	4	3
(2853, 3363)	3.7680	4	4
(3067, 703)	1.0212	2	1
(3317, 3793)	4.1885	4	4
(3727, 2259)	2.2826	2	2
(4796, 2761)	4.1454	4	4
(5451, 969)	4.4651	4	5

Outline

Summary

In this project, we implement ALS, SIALS to do the matrix factorization. We use the direct method and CG to solve ALS and use SVD to solve SIALS and then apply it to the recommender system.

Further Topics:

- Stochastic Gradient Descent (SGD)
- Nonnegative Matrix Factorization (NMF)

Reference

1. Trevor Hastie, Rahul Mazumder, Jason D. Lee and Reza Zadeh, Matrix Completion and Low-Rank SVD via Fast Alternating Least Squares, Journal of Machine Learning Research 16 (2015), 3367-3402.
2. Yehuda Koren, Robert Bell and Chris Volinsky, Matrix Factorization Techniques for Recommender Systems, *IEEE* Computer Society 42 (2009), 30-37.

THE END

Thanks for listening!