Big Data Mining Problems And Project Purpose (Use Python Language)

Problems that we may come across:

(a) Data missing problem: users cannot score every movies

(b) Features selection problem

(c) User vote frequency problem

Purpose of our project:

(a) Predict ratings a given user will give to a particular movie.

(b) Recommend similar movies to the target users

Data Sets:

MovieLens Dataset: http://grouplens.org/datasets/movielens/

Midterm: Matrix Factorization (1)

Data sets format: a lot of NA, simple user ID and movie ID

	Movie1	Movie2	Movie3	Movie4	Movie5	Movie6	Movie7	Movie8
User1	5	3	NA	NA	NA	NA	1	NA
User2	NA	NA	NA	4	1	NA	NA	NA
User3	NA	NA	2	NA	NA	NA	3	NA
User4	NA	NA	NA	NA	4	NA	NA	2
User5	NA	1	NA	3	NA	5	NA	NA

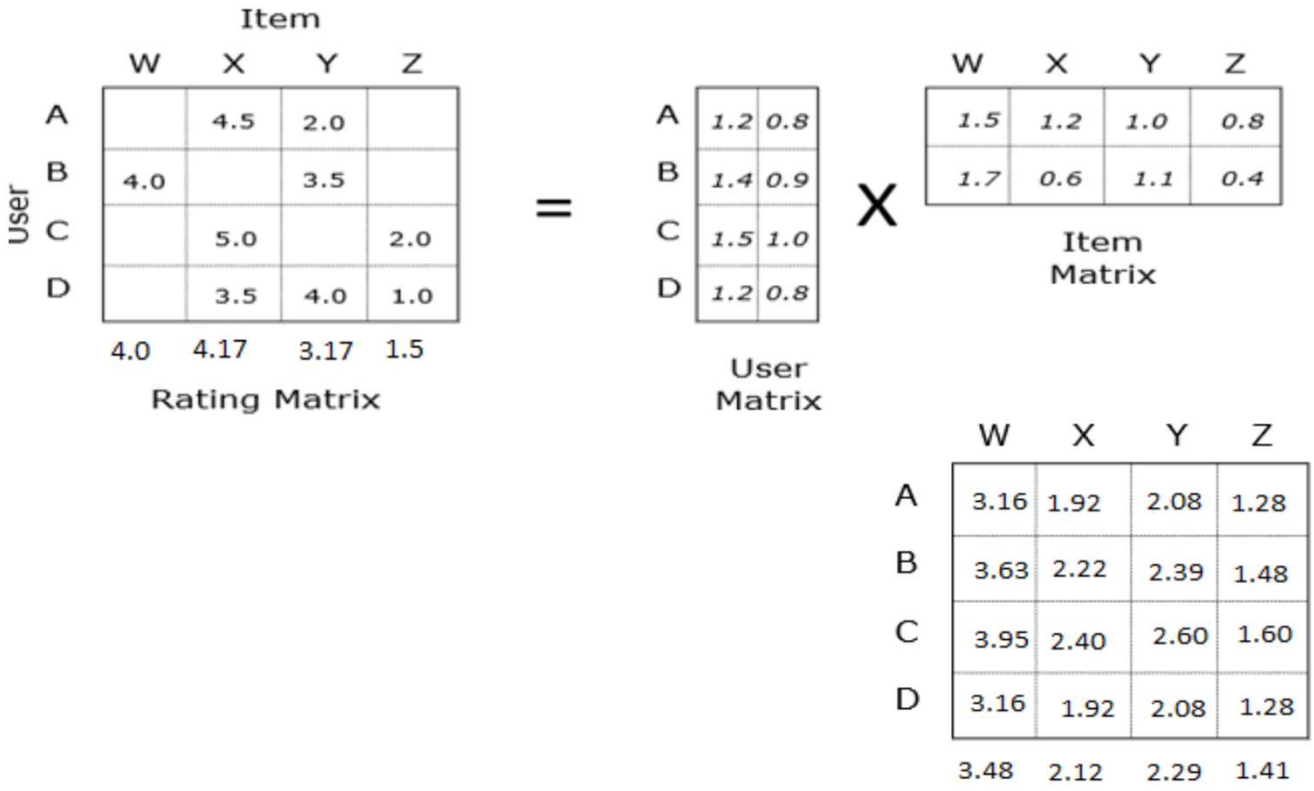
Midterm: Matrix Factorization (2)

$$\mathbf{R} \approx \mathbf{P} \times \mathbf{Q}^T = \hat{\mathbf{R}}$$

$\mathbf{P}$ (a $|U| \times K$ matrix)

$\mathbf{Q}$ (a $|D| \times K$ matrix)

K latent features



Midterm: Matrix Factorization (3)

$$\hat{r}_{ij} = p_i^T q_j = \sum_{k=1}^{k} p_{ik} q_{kj}$$

$$e_{ij}^2 = (r_{ij} - \hat{r}_{ij})^2 = (r_{ij} - \sum_{k=1}^{K} p_{ik} q_{kj})^2$$

$$\frac{\partial}{\partial p_{ik}} e_{ij}^2 = -2(r_{ij} - \hat{r}_{ij})(q_{kj}) = -2e_{ij} q_{kj}$$
$$\frac{\partial}{\partial q_{ik}} e_{ij}^2 = -2(r_{ij} - \hat{r}_{ij})(p_{ik}) = -2e_{ij} p_{ik}$$

$$p'_{ik} = p_{ik} + \alpha \frac{\partial}{\partial p_{ik}} e_{ij}^2 = p_{ik} + 2\alpha e_{ij} q_{kj}$$
$$q'_{kj} = q_{kj} + \alpha \frac{\partial}{\partial q_{kj}} e_{ij}^2 = q_{kj} + 2\alpha e_{ij} p_{ik}$$

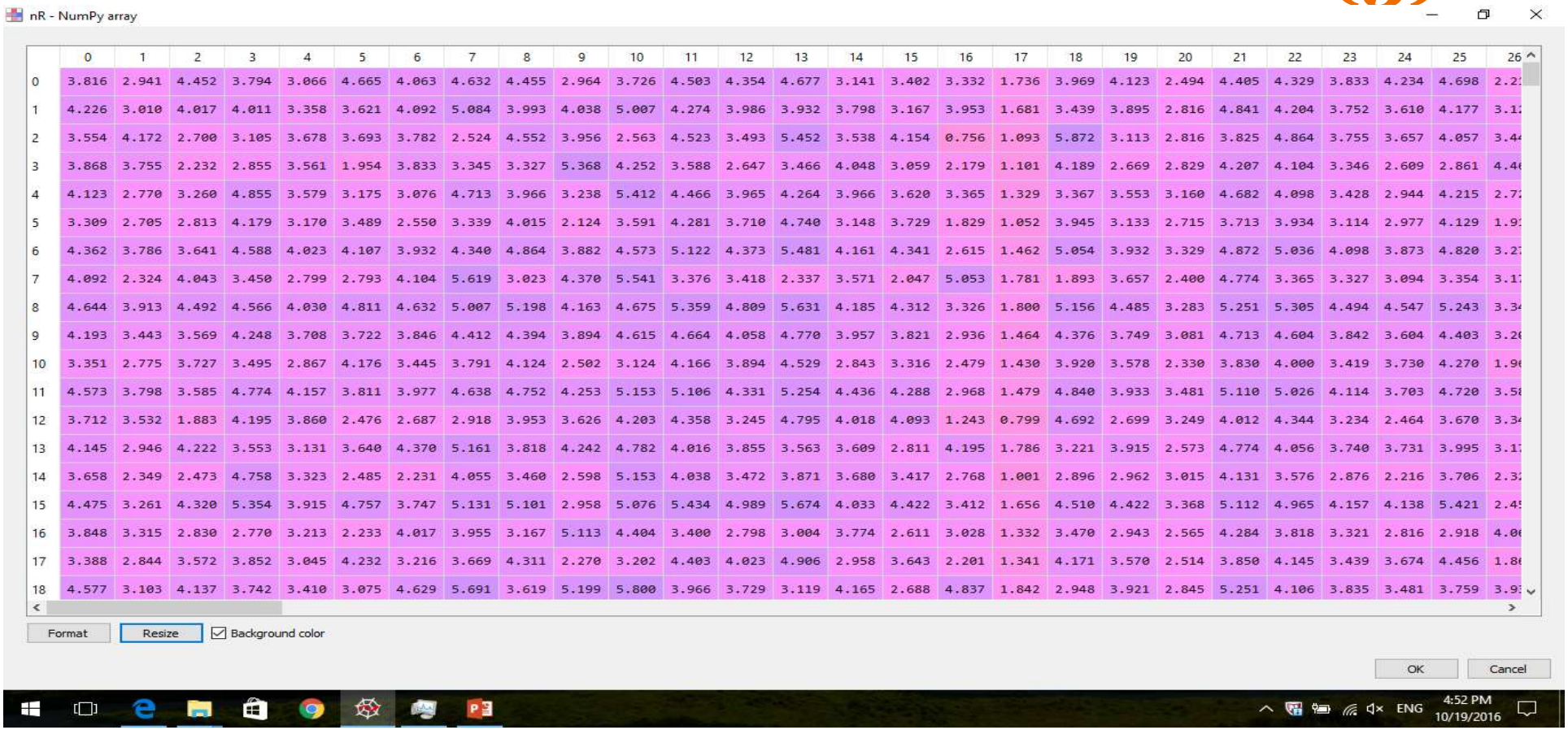
programming formula

regularization to avoid overfitting adding a parameter β

$$e_{ij}^2 = (r_{ij} - \sum_{k=1}^{K} p_{ik} q_{kj})^2 + \frac{\beta}{2} \sum_{k=1}^{K} (||P||^2 + ||Q||^2)$$
$$p'_{ik} = p_{ik} + \alpha \frac{\partial}{\partial p_{ik}} e_{ij}^2 = p_{ik} + \alpha(2e_{ij} q_{kj} - \beta p_{ik})$$
$$q'_{kj} = q_{kj} + \alpha \frac{\partial}{\partial q_{kj}} e_{ij}^2 = q_{kj} + \alpha(2e_{ij} p_{ik} - \beta q_{kj})$$

Midterm: Matrix Factorization (4)

Midterm: Matrix Factorization (5)



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
0	3.816	2.941	4.452	3.794	3.066	4.665	4.063	4.632	4.455	2.964	3.726	4.503	4.354	4.677	3.141	3.402	3.332	1.736	3.969	4.123	2.494	4.405	4.329	3.833	4.234	4.698	2.2
1	4.226	3.010	4.017	4.011	3.358	3.621	4.092	5.084	3.993	4.038	5.007	4.274	3.986	3.932	3.798	3.167	3.953	1.681	3.439	3.895	2.816	4.841	4.204	3.752	3.610	4.177	3.1
2	3.554	4.172	2.700	3.105	3.678	3.693	3.782	2.524	4.552	3.956	2.563	4.523	3.493	5.452	3.538	4.154	0.756	1.093	5.872	3.113	2.816	3.825	4.864	3.755	3.657	4.057	3.4
3	3.868	3.755	2.232	2.855	3.561	1.954	3.833	3.345	3.327	5.368	4.252	3.588	2.647	3.466	4.048	3.059	2.179	1.101	4.189	2.669	2.829	4.207	4.104	3.346	2.609	2.861	4.4
4	4.123	2.770	3.260	4.855	3.579	3.175	3.076	4.713	3.966	3.238	5.412	4.466	3.965	4.264	3.966	3.620	3.365	1.329	3.367	3.553	3.160	4.682	4.098	3.428	2.944	4.215	2.7
5	3.309	2.705	2.813	4.179	3.170	3.489	2.550	3.339	4.015	2.124	3.591	4.281	3.710	4.740	3.148	3.729	1.829	1.052	3.945	3.133	2.715	3.713	3.934	3.114	2.977	4.129	1.9
6	4.362	3.786	3.641	4.588	4.023	4.107	3.932	4.340	4.864	3.882	4.573	5.122	4.373	5.481	4.161	4.341	2.615	1.462	5.054	3.932	3.329	4.872	5.036	4.098	3.873	4.820	3.2
7	4.092	2.324	4.043	3.450	2.799	2.793	4.104	5.619	3.023	4.370	5.541	3.376	3.418	2.337	3.571	2.047	5.053	1.781	1.893	3.657	2.400	4.774	3.365	3.327	3.094	3.354	3.1
8	4.644	3.913	4.492	4.566	4.030	4.811	4.632	5.007	5.198	4.163	4.675	5.359	4.809	5.631	4.185	4.312	3.326	1.800	5.156	4.485	3.283	5.251	5.305	4.494	4.547	5.243	3.3
9	4.193	3.443	3.569	4.248	3.708	3.722	3.846	4.412	4.394	3.894	4.615	4.664	4.058	4.770	3.957	3.821	2.936	1.464	4.376	3.749	3.081	4.713	4.604	3.842	3.604	4.403	3.2
10	3.351	2.775	3.727	3.495	2.867	4.176	3.445	3.791	4.124	2.502	3.124	4.166	3.894	4.529	2.843	3.316	2.479	1.430	3.920	3.578	2.330	3.830	4.000	3.419	3.730	4.270	1.9
11	4.573	3.798	3.585	4.774	4.157	3.811	3.977	4.638	4.752	4.253	5.153	5.106	4.331	5.254	4.436	4.288	2.968	1.479	4.840	3.933	3.481	5.110	5.026	4.114	3.703	4.720	3.5
12	3.712	3.532	1.883	4.195	3.860	2.476	2.687	2.918	3.953	3.626	4.203	4.358	3.245	4.795	4.018	4.093	1.243	0.799	4.692	2.699	3.249	4.012	4.344	3.234	2.464	3.670	3.3
13	4.145	2.946	4.222	3.553	3.131	3.640	4.370	5.161	3.818	4.242	4.782	4.016	3.855	3.563	3.609	2.811	4.195	1.786	3.221	3.915	2.573	4.774	4.056	3.740	3.731	3.995	3.1
14	3.658	2.349	2.473	4.758	3.323	2.485	2.231	4.055	3.460	2.598	5.153	4.038	3.472	3.871	3.680	3.417	2.768	1.001	2.896	2.962	3.015	4.131	3.576	2.876	2.216	3.706	2.3
15	4.475	3.261	4.320	5.354	3.915	4.757	3.747	5.131	5.101	2.958	5.076	5.434	4.989	5.674	4.033	4.422	3.412	1.656	4.510	4.422	3.368	5.112	4.965	4.157	4.138	5.421	2.4
16	3.848	3.315	2.830	2.770	3.213	2.233	4.017	3.955	3.167	5.113	4.404	3.400	2.798	3.004	3.774	2.611	3.028	1.332	3.470	2.943	2.565	4.284	3.818	3.321	2.816	2.918	4.0
17	3.388	2.844	3.572	3.852	3.045	4.232	3.216	3.669	4.311	2.270	3.202	4.403	4.023	4.906	2.958	3.643	2.201	1.341	4.171	3.570	2.514	3.850	4.145	3.439	3.674	4.456	1.8
18	4.577	3.103	4.137	3.742	3.410	3.075	4.629	5.691	3.619	5.199	5.800	3.966	3.729	3.119	4.165	2.688	4.837	1.842	2.948	3.921	2.845	5.251	4.106	3.835	3.481	3.759	3.9

Midterm: Matrix Factorization (6)

Next Step:

(1) 10K→ 100K → 20M (newest)

(2) RMSE (accuracy), Bias-Variance tradeoff (K, beta)

(3) Recommend movie for users (Column pair RMSE)

(4) Timeline of Oscar Prediction: Best Director

Method	K=5	K=10
CF	0.911	0.911
BaseMF	0.878	0.863
STE	0.864	0.852
SocialMF	0.821	0.815

Table 3: RMSE values for comparison partners on Flixster with different settings of dimensionality K.

Figures 6 and 7 compare the RMSE of our model for different ranges of values for λ_T in both data sets. As shown in these figures, SocialMF has its best results on Epinions for $\lambda_T = 5$, and $\lambda_T = 1$ for Flixster.

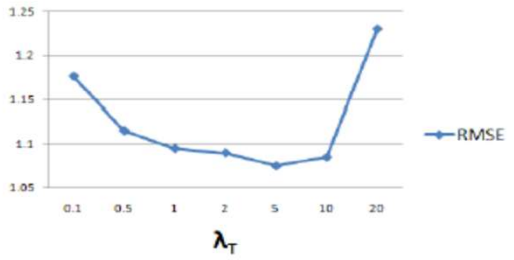
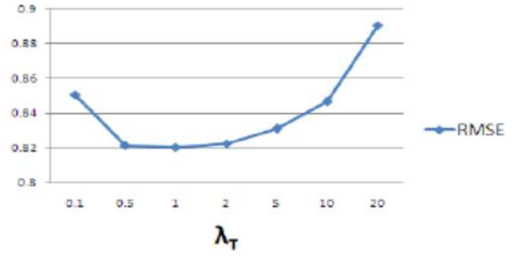
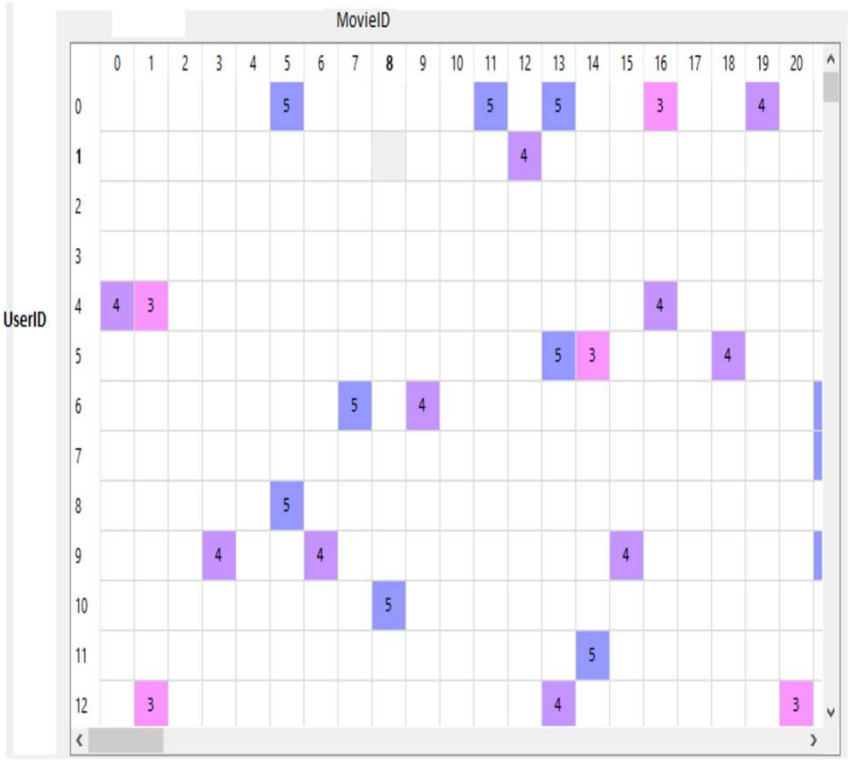


Figure 6: Impact of different values of λ_T on the performance of prediction in Epinions.

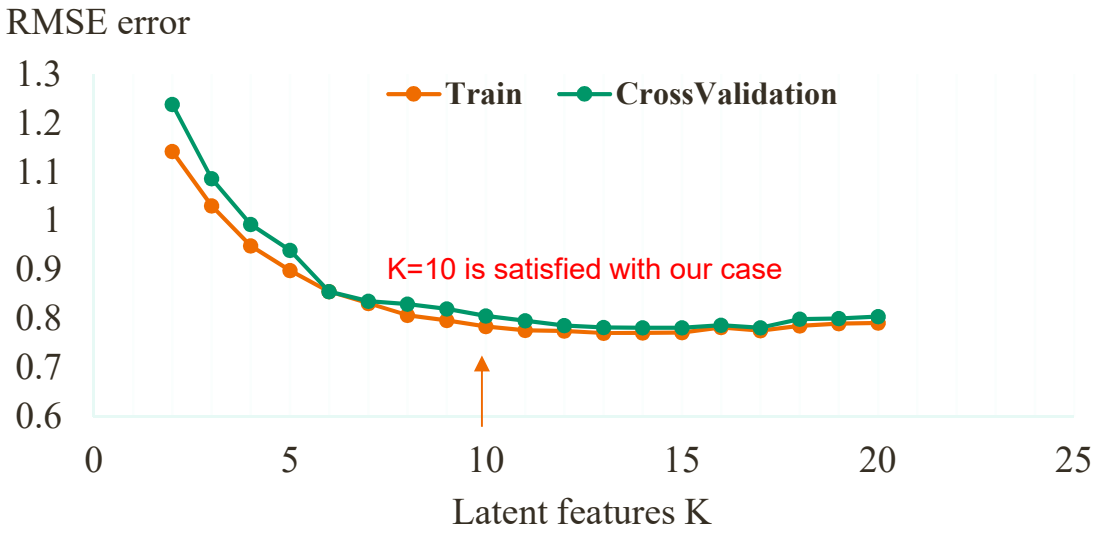


Solve The Missing Data Sets Problem (Original Data: Huge Missing Data Sets)
Use K (K=10) Latent Features Following User Habits (Average RMSE=0.8056)



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	3.829	3.640	3.257	3.757	4.040	4.970	3.822	5.783	5.087	2.024	4.633	4.574	3.009	5.070	3.068
1	4.255	3.337	3.718	3.332	3.828	3.655	4.061	3.961	3.337	3.416	3.648	4.029	3.920	3.966	4.467
2	4.212	3.979	3.868	1.362	3.477	2.773	2.787	1.873	2.973	3.686	2.042	3.288	3.671	4.624	4.030
3	4.298	3.974	3.828	3.202	4.784	3.529	4.619	3.995	5.172	2.816	5.598	5.567	4.633	4.296	4.314
4	3.098	2.986	2.861	2.246	3.384	4.477	1.718	4.284	4.701	3.649	4.210	3.820	4.402	4.473	3.059
5	2.590	3.680	3.462	1.621	3.076	4.843	1.745	3.846	5.396	4.788	4.984	4.849	3.198	4.397	2.564
6	3.914	4.134	3.436	4.946	4.280	4.873	4.004	5.425	4.396	3.929	5.472	4.551	3.836	5.262	4.611
7	3.636	2.672	3.043	3.419	3.354	4.595	2.429	4.604	3.526	3.189	2.681	3.708	3.095	6.409	2.549
8	4.796	4.645	4.609	3.865	4.769	4.941	4.785	5.283	5.248	3.463	5.274	5.468	4.505	5.555	4.795
9	4.330	3.836	3.707	4.079	4.119	3.710	4.820	4.221	3.872	3.442	4.770	4.458	3.781	4.618	4.511
10	3.709	3.478	4.012	1.397	2.968	3.582	3.887	3.354	4.600	2.532	3.776	4.197	3.227	4.588	2.570
11	5.354	3.932	4.232	4.959	3.985	3.638	6.726	4.544	3.226	3.622	4.505	4.097	3.387	5.210	4.909
12	2.982	3.050	2.866	1.714	4.152	3.864	1.983	3.465	6.069	3.567	5.638	5.584	4.778	4.740	2.430

Bias-Variance Tradeoff (steps=50)



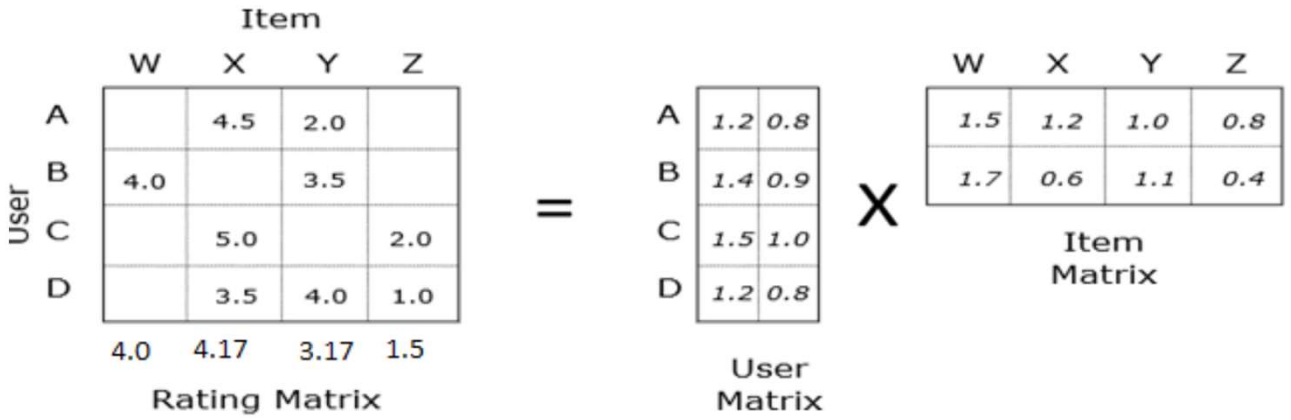
Matrix Factorization Algorithm (No.1 Algorithm In Open Netflix Competition)

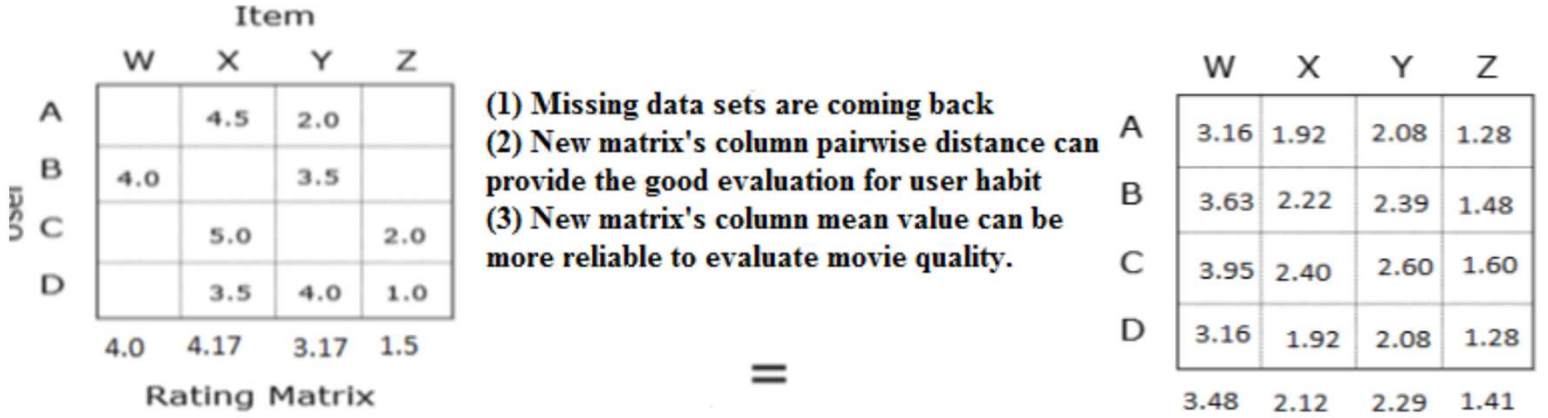
$$\mathbf{R} \approx \mathbf{P} \times \mathbf{Q}^T = \hat{\mathbf{R}}$$

$\mathbf{P}$ (a $|U| \times K$ matrix)

$\mathbf{Q}$ (a $|D| \times K$ matrix)

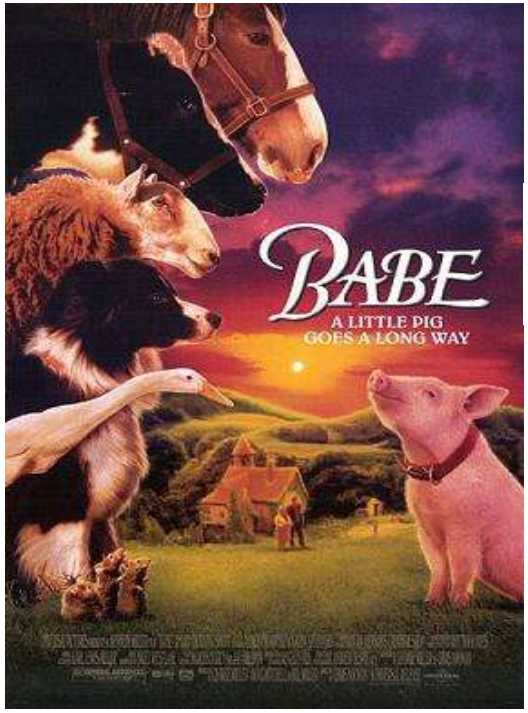
K latent features





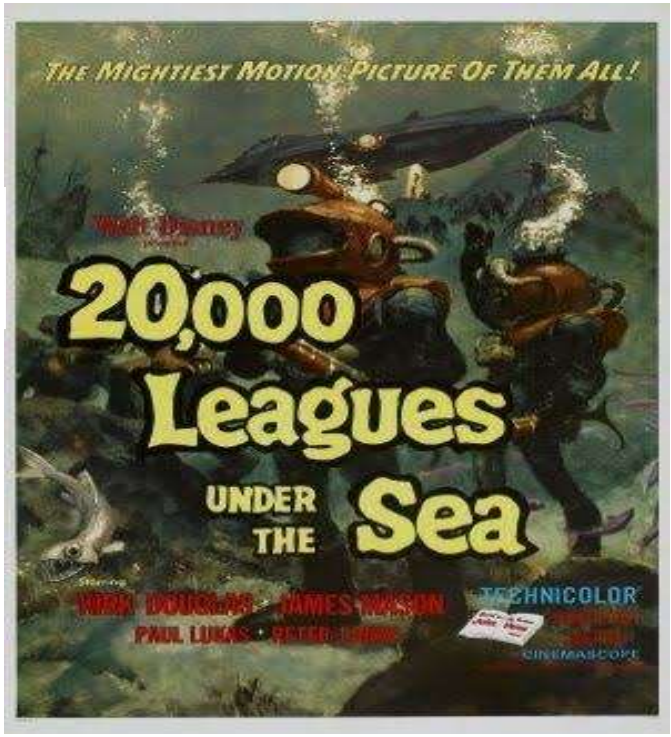
Automated Precisely Recommend Target Movies For Users (min distance =1.38447 > average 0.8056)

Babe is a 1995 Australian-American comedy-drama film



Pair Distance is 1.38447 By K (K=10) latent features

20,000 Leagues Under the Sea is a 1954 American Technicolor adventure film



Automated Recommend Movies to Precisely Target Customers With Lower Cost

Crimson Tide Min Pair Distance = 0.20926 with The Specialist

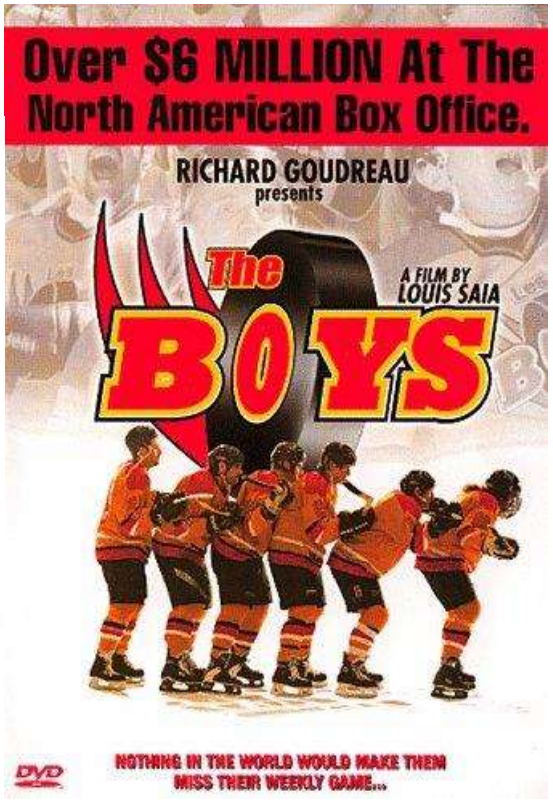
Crimson Tide is a 1995 American submarine film

Pair Distance is 0.20926 By K (K=10) latent features

The Specialist is a 1994 American action film

Automated Recommend the Higher Quality Movies to Precisely Target Customers
Les Boys Receives the Highest Ratings Twice

Timeline	MovieID	Movie Title	Top 3 MF Score	Prize
10K	1242	Old Lady Who Walked in the Sea	5.286	1992 César Award for Best Actress
	835	The Gay Divorcee	5.033	nominated for the Academy Award for Best Picture in 1934
	641	Paths of Glory	4.998	nominated for a BAFTA Award under the category Best Film
20K	1463	Les Boys	5.341	It has spawned three sequels and by any measure (profit, box office or attendance)
	1591	Fallen Angels	5.237	32th Golden Horse Awards
	884	Year of the Horse	5.045	
30K	1463	Les Boys	5.166	It has spawned three sequels and by any measure (profit, box office or attendance)
	1554	Safe Passage	5.014	
	1175	Hugo Dugay	4.943	



Recommendation Movies Using Machine Learning Methods

Summary:

(1) Matrix factorization is a great **machine learning technology** to solve the missing information problem (No 1 algorithm in Netflix Prize open competition).

(2) It can be used for online **big data mining**, like movies rating, financial products survey etc.

(3) The results can be used to recommend various products to **precisely target customers with very lower fees**.

(4) It also have both industry and academic impacts to monitor **the social consuming habit changes**.