Movie recommendation: Collaborative Filtering

Vishal Anand (UNI: va2361)
Department of Computer Science, Columbia University
New York City, NY, USA
vishal.anand@columbia.edu

***Abstract*—As part of homework 1 of COMS-6998 : Advanced Machine Learning with Personalization, a movie recommendation engine is built which performs collaborative filtering. The rank matrices are factorized and completed by decomposing through low-rank factorization.**

I. Introduction (*Heading 1*)

We are working on the MovieLens 20M dataset which contains 20000263 ratings across 27278 movies as generated by 138493 users between January 09, 1995 to March 31, 2015. All selected users have rated at least 20 movies. The file 'ratings.csv' contains the ratings given to some movies on a 5-star scale with half-increments. On each line, the file has a rating with the following format :
(userId; movieId; rating; timestamp).

II. Approach

We have 80 data-outputs for each of these configurations:

- Ranks = [10, 20, 30, 60]
- Lambda = [0.001, 0.02, 0.1, 1.0]
- Splits = [split_1, split_2, split_3, split_4, split_5]

Data produced for Rank=10 on Split1 is displayed at the end (all the data is placed at https://github.com/vishalanand/Advanced-ML-Product-Ranking):

Parts C, D, E and F are described in the algorithm section and in the code

A. Creating multiple data-splits

We created randomized index based split for each user and the training and testing were carried out based on SGD on perceptron based updates per user and movie input data point

B. Loading the data-set

The data-set is read as a text file and the elements are loaded per entry-wise. Such as, we iterate over each data-point

C. Matrix Update and Stochastic Gradient Descent

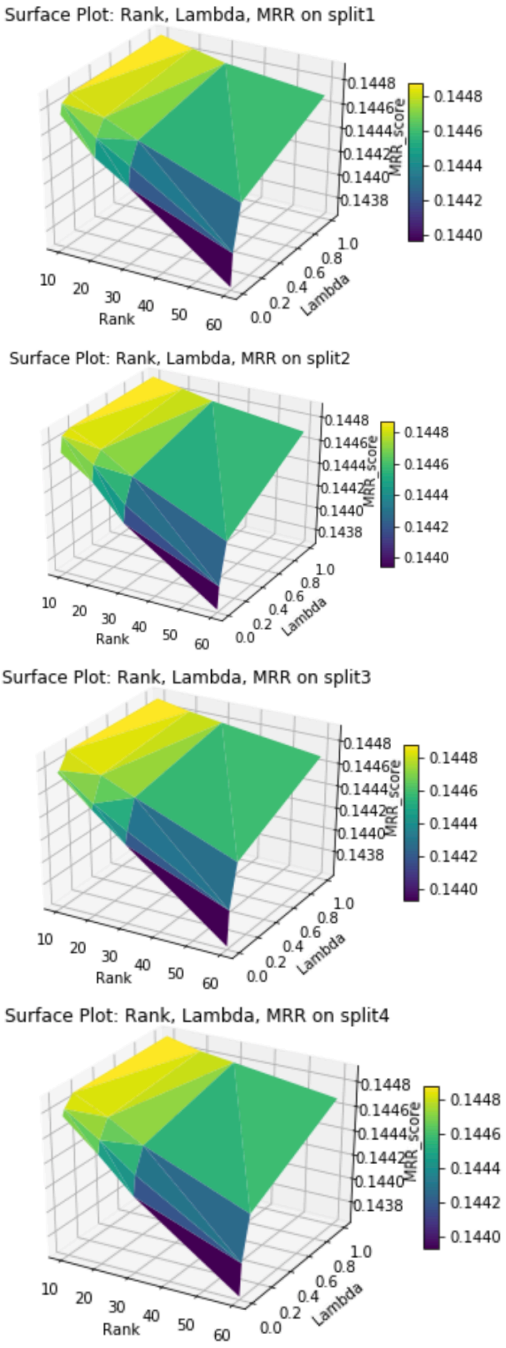
D. Root Mean Square Error Calculation

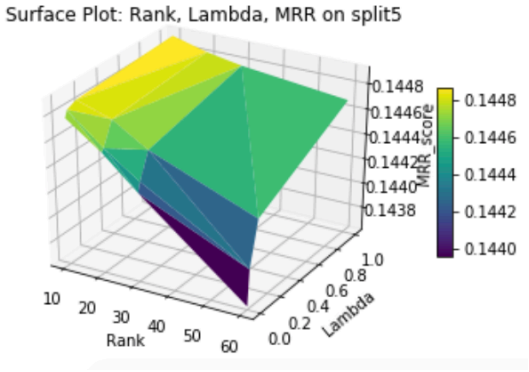
E. Mean Reciprocal Rank Calculation

F. Grid Search

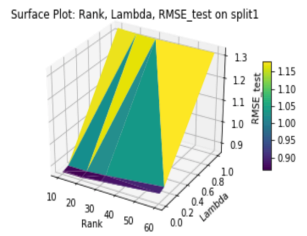
III. Graphs

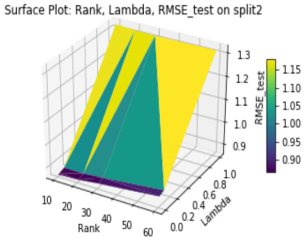
A. Surface-Plots of MRR:

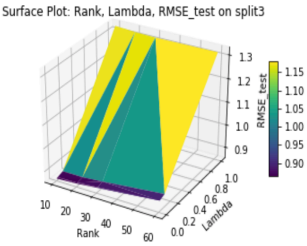


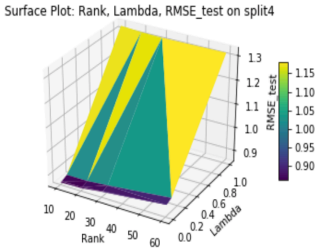


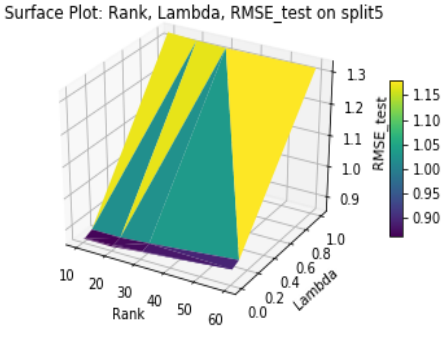
B. *Surface-Plots of RMSE_test*



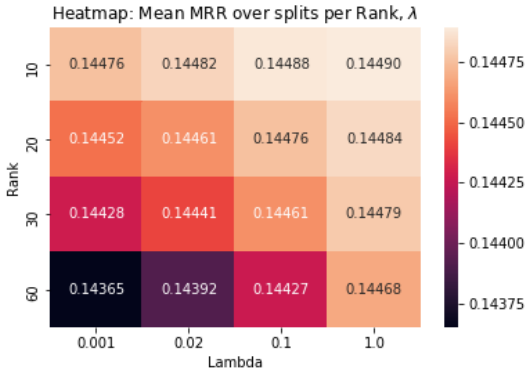


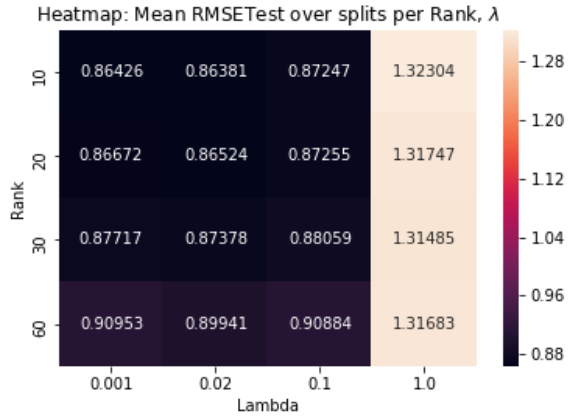






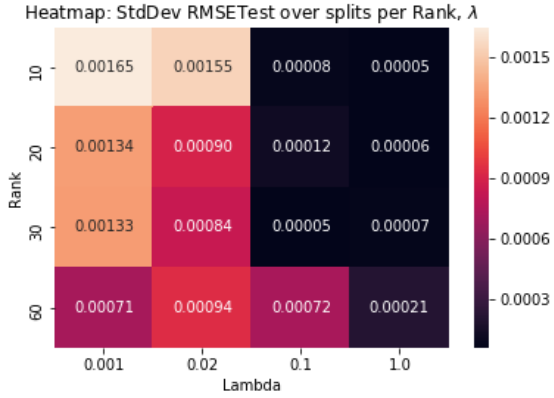
C. *Heat-maps Mean MRR, RMSE_test over splits:*

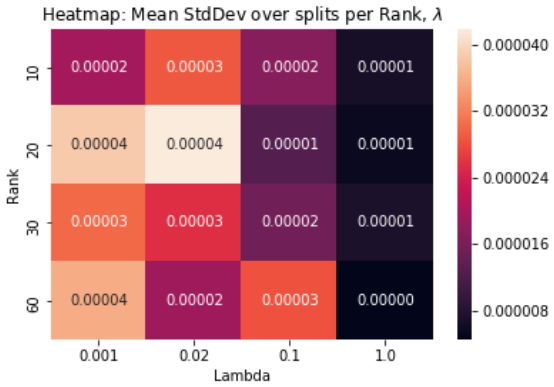




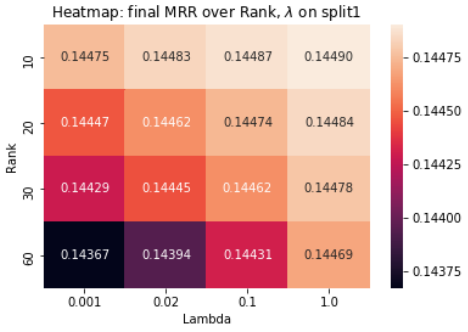
D.

E. *Heat-maps StdDev MRR, RMSE_test over splits: StdDev over RMSE_Test(image1), MRR(image2)*

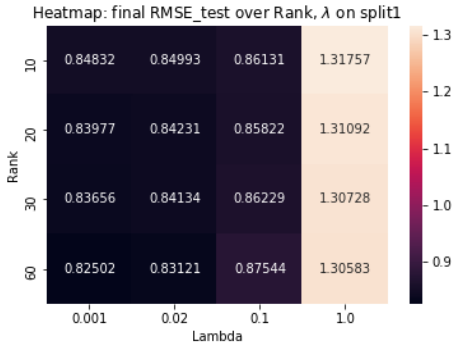




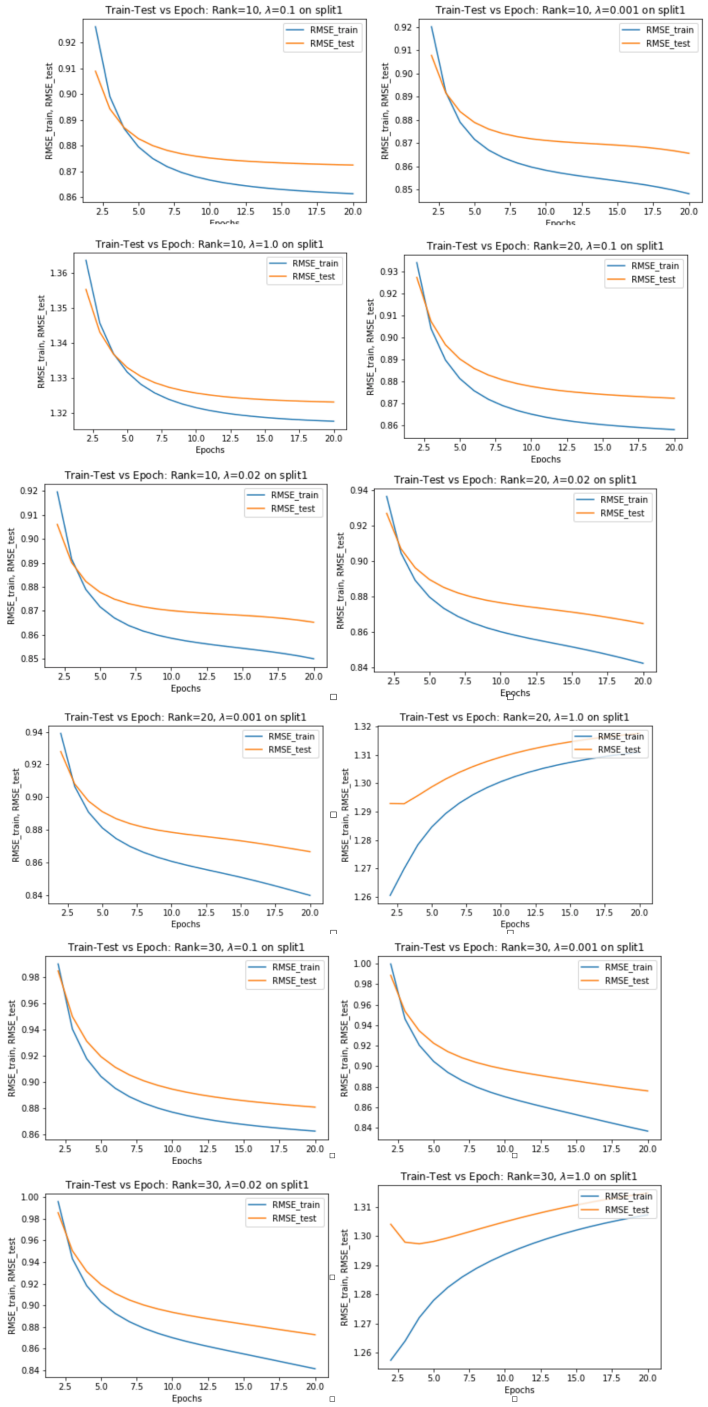
F. Heat-Maps for RMSE_Test on split_1

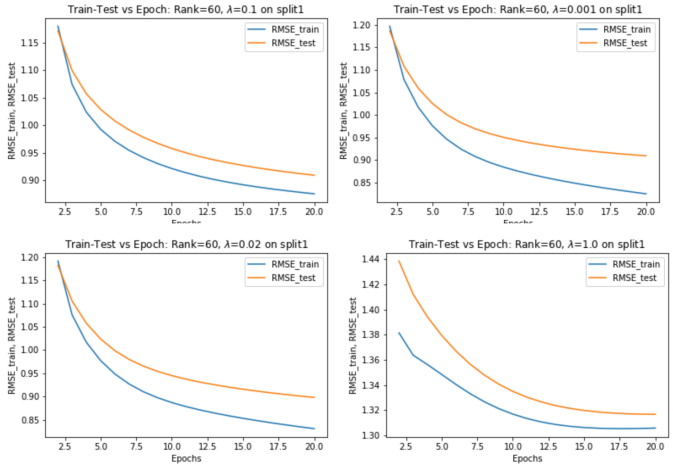


G. Heat-Maps for MRR on split_1



H. RMSE_train, RMSE_test and Epochs





IV. MRR VS EPOCHS

V. ALGORITHM

$$\underset{\mathbf{w},\mathbf{h}}{\operatorname{argmin}} \sum_{(u,i)\in\mathcal{Z}} (v_{ui} - \mathbf{w}_u^T \mathbf{h}_i)^2 + \lambda\left(\sum_i ||\mathbf{w}_i||^2 + \sum_u ||\mathbf{h}_u||^2\right)$$

$$e_{ui} \leftarrow v_{ui} - \mathbf{w}_u^T \mathbf{h}_i$$

$$\mathbf{w}_u \leftarrow \mathbf{w}_u + \gamma(e_{ui}\mathbf{h}_i - \lambda \mathbf{w}_u)$$

$$\mathbf{h}_i \leftarrow \mathbf{h}_i + \gamma(e_{ui}\mathbf{w}_u - \lambda \mathbf{h}_i)$$

The core logic is mentioned in the code here:
https://github.com/vishalanand/Advanced-ML-Product-Ranking/blob/master/final.py#L54

```
V[user, :] = V[user, :] + eta * ((error * W[movie,:]) - (lambd * V[user,:]))
W[movie, :] = W[movie, :] + eta * ((error * V[user,:]) - (lambd * W[movie,:]))
```

VI. RMSE-TEST, RMSE-TRAIN, MRR DATA PRODUCED

A. Data generated for Split-1, Rank=10

The data is described for split=1, Rank=10 over different lambda values. All the rest of the information is located in https://github.com/vishalanand/Advanced-ML-Product-Ranking.

We have 80 data-outputs for each of these configurations:
- Ranks = [10, 20, 30, 60]
- Lambda = [0.001, 0.02, 0.1, 1.0]
- Splits = [split_1, split_2, split_3, split_4, split_5]

Rank	Lambda	Iter	RMSE_train	RMSE_test	MRR
10	0.001	1	1.038286	0.953560	0.144645
10	0.001	2	0.920051	0.907733	0.144786
10	0.001	3	0.891892	0.891609	0.144810
10	0.001	4	0.879021	0.883577	0.144815
10	0.001	5	0.871695	0.878953	0.144821
10	0.001	6	0.867026	0.876059	0.144823
10	0.001	7	0.863824	0.874144	0.144818
10	0.001	8	0.861506	0.872826	0.144818
10	0.001	9	0.859752	0.871889	0.144813
10	0.001	10	0.858373	0.871202	0.144809
10	0.001	11	0.857244	0.870679	0.144803
10	0.001	12	0.856283	0.870260	0.144797
10	0.001	13	0.855425	0.869898	0.144789
10	0.001	14	0.854620	0.869553	0.144789
10	0.001	15	0.853821	0.869186	0.144783
10	0.001	16	0.852982	0.868757	0.144765
10	0.001	17	0.852056	0.868226	0.144768
10	0.001	18	0.850996	0.867553	0.144765
10	0.001	19	0.849760	0.866707	0.144755
10	0.001	20	0.848324	0.865678	0.144751

TABLE I
SPLIT1: RANK=10, λ=0.001

Rank	Lambda	Iter	RMSE_train	RMSE_test	MRR
10	0.02	1	1.039179	0.950775	0.144744
10	0.02	2	0.919648	0.906044	0.144821
10	0.02	3	0.891683	0.890161	0.144827
10	0.02	4	0.878897	0.882247	0.144842
10	0.02	5	0.871627	0.877696	0.144857
10	0.02	6	0.867002	0.874848	0.144851
10	0.02	7	0.863839	0.872963	0.144856
10	0.02	8	0.861558	0.871662	0.144858
10	0.02	9	0.859842	0.870732	0.144851
10	0.02	10	0.858503	0.870045	0.144842
10	0.02	11	0.857419	0.869519	0.144840
10	0.02	12	0.856511	0.869096	0.144841
10	0.02	13	0.855717	0.868735	0.144840
10	0.02	14	0.854993	0.868401	0.144841
10	0.02	15	0.854298	0.868062	0.144843
10	0.02	16	0.853594	0.867686	0.144841
10	0.02	17	0.852843	0.867242	0.144840
10	0.02	18	0.852007	0.866695	0.144831
10	0.02	19	0.851046	0.866013	0.144829
10	0.02	20	0.849927	0.865172	0.144833

TABLE II
SPLIT1: RANK=10, λ=0.02

Rank	Lambda	Iter	RMSE_train	RMSE_test	MRR
10	0.1	1	1.045042	0.949757	0.144772
10	0.1	2	0.926225	0.908935	0.144856
10	0.1	3	0.899134	0.894323	0.144871
10	0.1	4	0.886671	0.886996	0.144871
10	0.1	5	0.879550	0.882727	0.144867
10	0.1	6	0.874995	0.880008	0.144857
10	0.1	7	0.871867	0.878166	0.144867
10	0.1	8	0.869608	0.87686	0.144873
10	0.1	9	0.867914	0.875904	0.144870
10	0.1	10	0.866605	0.875182	0.144871
10	0.1	11	0.865568	0.874622	0.144871
10	0.1	12	0.864731	0.874180	0.144871
10	0.1	13	0.864041	0.873822	0.144869
10	0.1	14	0.863465	0.873529	0.144866
10	0.1	15	0.862976	0.873283	0.144866
10	0.1	16	0.862555	0.873074	0.144866
10	0.1	17	0.862189	0.872892	0.144865
10	0.1	18	0.861866	0.872731	0.144867
10	0.1	19	0.861576	0.872585	0.144870
10	0.1	20	0.861314	0.872451	0.144873

TABLE III
SPLIT1: RANK=10, λ=0.1

Rank	Lambda	Iter	RMSE_train	RMSE_test	MRR
10	1.0	1	1.420639	1.389550	0.144637
10	1.0	2	1.363616	1.355283	0.144778
10	1.0	3	1.345713	1.343128	0.144826
10	1.0	4	1.336905	1.336768	0.144846
10	1.0	5	1.331614	1.332913	0.144857
10	1.0	6	1.328114	1.330373	0.144871
10	1.0	7	1.325657	1.328603	0.144878
10	1.0	8	1.323859	1.327322	0.144885
10	1.0	9	1.322504	1.326367	0.144888
10	1.0	10	1.321457	1.325638	0.144891
10	1.0	11	1.320635	1.325073	0.144895
10	1.0	12	1.319979	1.324628	0.144895
10	1.0	13	1.319448	1.324273	0.144899
10	1.0	14	1.319015	1.323987	0.144901
10	1.0	15	1.318658	1.323754	0.144901
10	1.0	16	1.318361	1.323564	0.144901
10	1.0	17	1.318112	1.323407	0.144904
10	1.0	18	1.317903	1.323277	0.144902
10	1.0	19	1.317725	1.323169	0.144905
10	1.0	20	1.317574	1.323078	0.144904

TABLE IV
SPLIT1: RANK=10, λ=1.0

REFERENCES

1. https://www.cs.cmu.edu/~mgormley/courses/10601-s17/slides/lecture25-mf.pdf
2. https://github.com/vishalanand/Advanced-ML-Product-Ranking
3. http://www.cs.columbia.edu/~jebara/6998/Notes1.pdf