ISyE 6740 – Spring 2021
Project Proposal

Team Member Names: Christopher Dae-Hyun Kim @ ckim612

Project Title: Association Rules as Sparsity Alleviation in Recommendation Systems

Problem Statement
Netflix held an open competition in 2007 to teams that managed to beat their in-house recommendation system CineMatch by 10% using the root mean squared error (RMSE) as the evaluation metric.[1] Sophistication of models varied greatly; some teams had futilely approached the problem by extracting features, such as genres, cast, etc., outside of the published dataset, some used a simple average based approach proved to be only marginally worse than CineMatch, and the winning teams submitted an ensemble of models.[1] The focus of this project will be to expand upon the single value decomposition (SVD) algorithmic approach by three students Michael Harris, Jeffrey Wang, and David Kamm [1] using association rules [2] as pseudo imputations to the sparse utility matrix.

Data Source
Netflix had recently republished the dataset that was used during the competition on Kaggle. The data consists of seven files with a total of 2.13 GB in size. According to the documentation, there are over 100,000,000 ratings from over 480,000 users over 17,000+ movie titles. Ratings are from October 1998 to December 2005. The team will also use the popular MovieLens 100k dataset from GroupLens. The data consists of 100,000 ratings from 1000 users on 1700 movie titles and was released on April 1998.

Data sources can be retrieved from the following links:
1. Kaggle: https://www.kaggle.com/netflix-inc/netflix-prize-data
2. MovieLens: https://grouplens.org/datasets/movielens/100k/

Due to lack of computational resources, an undetermined subset of the data from Netflix will be used whereas the entirety of the data from GroupLens will be used.

Methodology
Data will be processed as seen in the following figure:

	I_1	I_2	I_3	I_4
U_1	5	1	2	
U_2	3	3		3
U_3	5		3	
U_4				5

Figure 1. U_n indicates users and I_m indicates movie titles.

SVD is the factorization of a real matrix $M \in \mathbb{R}^{n\,x\,m}$ where $n \leq m$:

$$M = U\,\Sigma\,V^T := U\Sigma\Sigma^T V^T,$$
$$M = [u_1, u_2, \ldots u_n]\begin{bmatrix}\sigma_1^2 & & \\ & \ddots & \\ & & \sigma_n^2\end{bmatrix}[v_1, v_2, \ldots v_m]^T.$$

The objective function [3] is defined as minimizing the sum of squared errors (SSE):

$$\min_{U,V,\Sigma} \sum_{i,j \in M} (M_{ij} - [U\Sigma\Sigma^T V^T]_{ij})^2$$

Since SVD is undefined for a sparse utility matrix M [3] as presented in the data of users and their respective ratings for movies as seen in Figure 1, we must rewrite the SVD and the new objective function, with the introduction of regularization using stochastic gradient descent optimization [3,4], as:

$$R = Q^T \cdot P \text{ where } M \approx R,\ U\Sigma \approx P, \text{ and } \Sigma V^T \approx Q.$$

$$\min_{p,q} \sum_{u,i \in k} (r_{ui} - q_i^T p_u)^2 + \lambda[\sum_u \|p_u\|^2 + \sum_i \|q_i\|^2\ ]\ s.t.$$
$$\varepsilon_{ui} = r_{ui} - q_i^T p_u$$
$$q_i \leftarrow q_i + \gamma(\varepsilon_{ui}\, p_u - \lambda q_i)$$
$$p_i \leftarrow p_i + \gamma(\varepsilon_{ui}\, q_i - \lambda p_u)$$

This project will attempt to make the utility matrix less sparse by introducing averages dependent on association rule mining as "real" data. We will demonstrate this method explicitly by using Figure 1 as an example.

The first step would be to map the user ratings of movies as transactions: $U_1 = \{I_1, I_2, I_3\}, U_2 = \{I_1, I_2, I_4\}, U_3 = \{I_1, I_3\}, U_4 = \{I_4\}$. Suppose the item set $\{I_1, I_2\}$ is determined. The support of this item set would be $supp\{I_1, I_2\} = \frac{2}{4}$. The confidence of the rule would be $conf\{I_1 \Rightarrow I_2\} = \frac{2}{3}$. The lift of the rule would be $lift\{I_1 \Rightarrow I_2\} = \frac{8}{6}$.

Since $lift\{I_1 \Rightarrow I_2\} > 1$, the antecedent and consequent are dependent of each other. We will then calculate the average for I_2 with respect to I_1 and then insert this value into the utility matrix as seen below in the figure:

	I_1	I_2	I_3	I_4
U_1	5	1	2	
U_2	3	3		3
U_3	5	2	3	
U_4				5

Figure 2. Utility matrix M with 2 inserted into $M_{3,2}$.

With this new matrix, we will then run the SVD algorithm as mentioned previously, which we will now refer to as the SVD-Alleviation (SVD-A) algorithm.

Evaluation

We will train the SVD-A model and the regular SVD model on 80% of data, and test on remaining 20%. We will train different models respective to each data source. We will report the RMSE for all models since the competition required this metric.

References

1. Leskovec J, Rajaraman A, Ullman J. Mining of Massive Datasets. 3rd ed. Cambridge University; 2020.

2. Agrawal R, Imieliński T, Swami A. Mining Association Rules between Sets of Items in Large Databases. ACM SIGMOD Record. 1993;22(2):207-216.

3. Leskovec J, Rajaraman A, Ullman J. Recommender Systems: Latent Factor Models [Slides]. CS246: Mining Massive Datasets, Stanford University.

4. Koren Y, Bell R, Volinsky C. Matrix Factorization Techniques for Recommender Systems. Computer. 2009; 42(8):30-37.