Project Report on Recommender System

Group-13

January 5, 2019

1 Motivation

Recommender systems are an intuitive line of defense against consumer over-choice. Given the explosive amount of available data, Matching consumers with the most appropriate products is key to enhance user satisfaction. Recommender Systems analyze patterns of user interest in products to provide personalized recommenda tions that suit a users taste. Because good personalized recommendations can add another dimension to the user experience, e-commerce leaders like Amazon.com and Netflix have made recommender systems a salient part of their websites.

2 Problem Description

When users to items ratings are low, ratings matrix for learning the model is very sparse. Sparseness of user-to-item ratings or reviews lowers the quality of Recommender systems. our aim is to improve the accuracy in case of sparse user-item interactions. According to basic Collaborative Filtering algorithm, items are recommended to users based on the ratings or reviews given by other users.However if a new item is added, it goes through the cold start phase due to the lack of valuable user interactions and the same holds for new user. In this project, We are trying to improve accuracy in case of sparse user-item interactions and tackle cold start problem.

3 Literature review and prior work

Recommendation models are usually classified into three categories collaborative filtering, content based and hybrid recommender system. Collaborative filtering makes recommendations by learning from user-item historical interactions, either explicit (e.g. users previous ratings) or implicit feedback (e.g. browsing history). Content-based recommendation is based primarily on comparisons across items and users auxiliary information. A diverse range of auxiliary information in form of texts, images and videos can be taken into account. For users, demographic information, age, sex etc can be taken into account and similarly for items, items features can be considered while learning. Hybrid model refers to recommender system that integrates two or more types of recommendation strategies. For cold start problem, content based recommender system is considered which takes into account the features of items and uses also with their interactions as ratings or reviews or clicks etc.

With recent advancement of deep leari Recent advances in deep learning based recommender systems have gained significant attention by overcoming obstacles of conventional models and achieving high recommendation quality. Deep learning is able to effectively capture the non-linear and non-trivial user-item relationships

4 Models

We consider following two different approaches as described below to tackle the problems.

1. **Matrix Factorization**

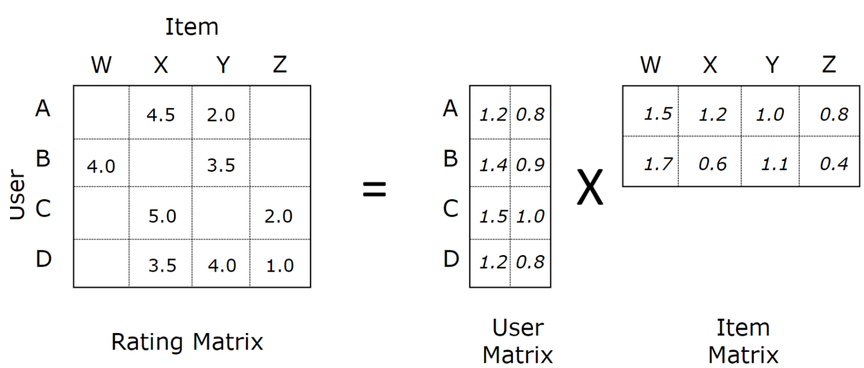

Figure 1: Low-Rank matrix completion

$$X = UV^T$$

where, Rating matrix X is a NxM partially observed matrix(sparse Matrix)(N is the number of users and M is the number of items). U is the NxK latent factor matrix where each of its row represents a K-dim latent factor u_n. V is the MxK latent factor matrix where each of its row represents a K-dim latent factor v_n. We need to approximate X as a product of two matrices, user matrix and item matrix, denoted by U and V respectively.
Loss function: $\sum_{(n,m)}(X_{nm} - u_n^T v_m)^2 + \sum_{n=1}^{N} \lambda_U ||u_n||^2 + \lambda_V ||v_m||^2$
After learning U and V, any missing X_{nm} can be approximated as $u_n^T v_m$

2. **Biased incoporating Matrix Factorisation** Bias is incorporated to include user's rating habits and item's average popularity. We add one column and one row for

items bias and user bias respectivelyvin latent matrix U and V. Initialization of Bias: user's bias is calculated by averaging the item's average ratings minus rating that he/she has rated over all the items.
Bias gets updated along with latent vectors of user and items during training the model.

3. **Autoencoders**
Autoencoder is an unsupervised model attempting to reconstruct its input data in the output layer. It compresses the input and uncompresses to reconstruct. An encoder (neural net) does compression and a decoder (neural net) does decompression.Cold start is equivalent to the missing data problem where preference information is missing.

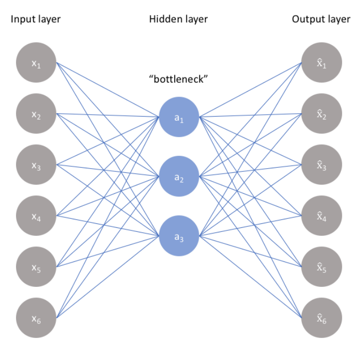

Figure 2: AutoRec

AutoRec, a new CF model based on the autoencoder paradigm; the idea of this paradigm stems from the recent successes of (deep) neural network models for vision and speech task.
We have used item-based autoencoder which take input as partially observed ratings of particular item by all users. This input is projected into a low-dimensional latent space and is then reconstructed in the output space to predict missing ratings.
The observed loss function is: $min_\theta \sum_{i=1}^{n} ||r^i - h(r^i, \theta)||_O^2 + \frac{\lambda}{2}(||W||_F^2 + ||V||_F^2)$

4. **Ensemble Model**

   We use predicted ratings of both models(Autoencoders and Matrix Factorisation) as features to train a new model and use the new model to make predictions on test data.

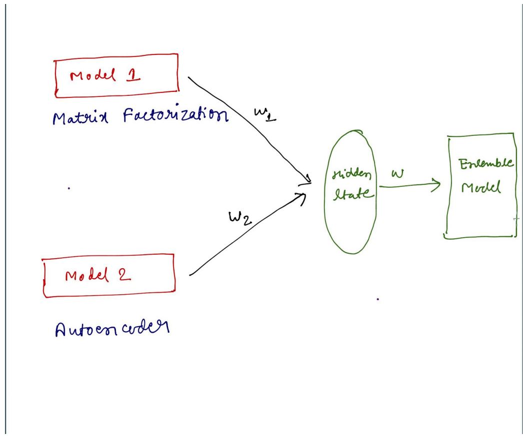

Figure 3: Ensemble Model

We train both Matrix factorization and Autoencoder model and compare RMSE among two and then combine results of both as input in ensemble model in order to improve accuracy.

5 Tools or Softwares used

We have implemented matrix completion code and modified code available for Autorec model[3] to use on our dataset and handle our considered problem of this project. We have done the coding implemetation in python using following libraries.

1. **Numpy**
2. **sklearn**
3. **pandas**
4. **scipy**
5. **pylab**
6. **Tensorflow**
   - AdamOptimizer
   - Exponential decay

6 Datset and Experimental Results

Dataset Used: Both of below stated datasets are very sparse.

- **Amazon Product Data** This dataset contains product reviews and metadata from Amazon, including 142.8 million reviews spanning May 1996 - July 2014.Ratings Only: Includes no metadata or reviews, but only (user,item,rating,timestamp) tuples.
- **MovieLens 100K Dataset** This data set consists of 100,000 ratings (1-5) from 943 users on 1682 movies.Each user has rated at least 20 movies.

6.1 Matrix Factorization

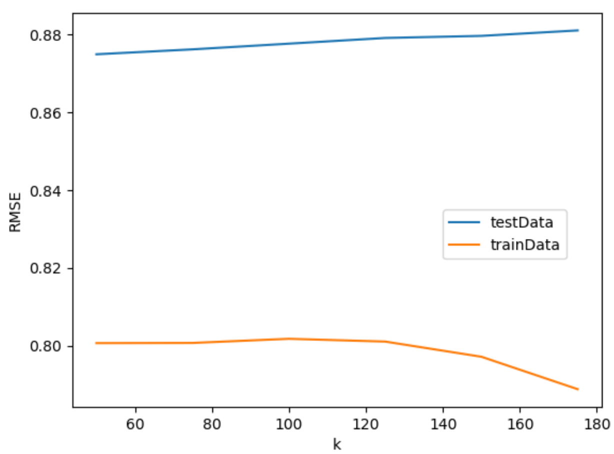

Figure 4: RMSE vs k

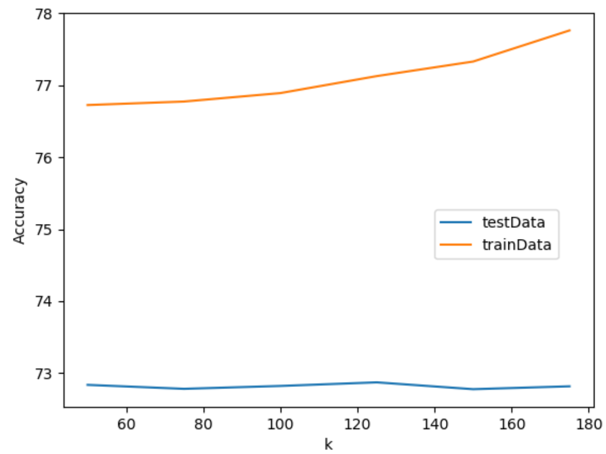

Figure 5: Accuracy vs k

6.2 Biased Matrix Factorization

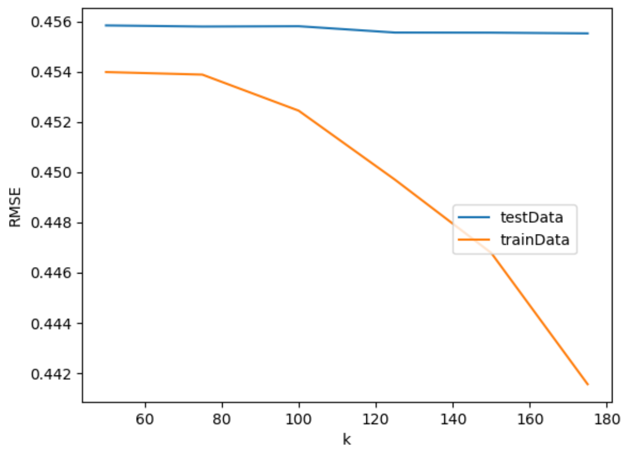

Figure 6: RMSE vs k

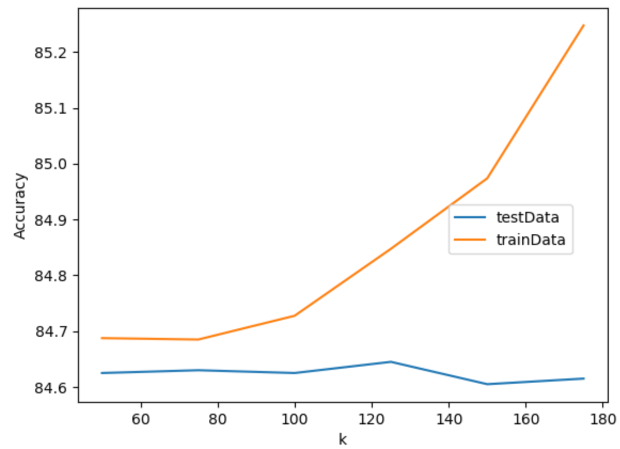

Figure 7: Accuracy vs k

Here, k is size of latent vector of each item and each user.

6.3 Autoencoder

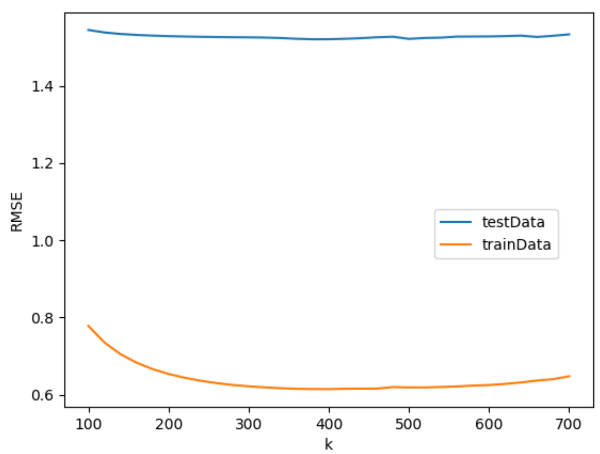

Figure 8: RMSE vs k

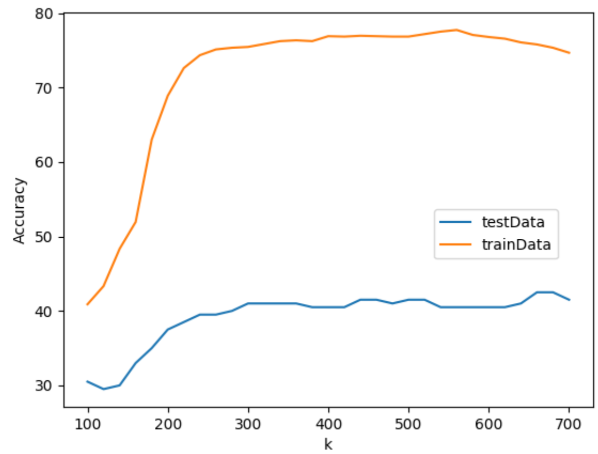

Figure 9: Accuracy vs k

Here, k is number of neurons in hidden layer.

6.4 Inductive Matrix Factorization

We have taken age,gender,sex of users as user feature vector from MovieLens dataset and genere of movie as item features.
Training accuracy: 68.1%
Test accuracy: 62.3%

6.5 Conclusion:

For matrix factorization:
Training accuracy: 75.7%
Test accuracy: 72.7%

For Biased matrix factorization:
Training accuracy: 84.5%
Test accuracy: 78.6%

For autoencoder:
Training accuracy: 73.1%
Test accuracy: 41.1%

For Ensemble model:
Training accuracy: 73.18%
Test accuracy: 46.25%
Training RMSE: 0.8005%
Test RMSE: 0.8751%

For Autoencoder model, we are not getting good accuracies on test error as we have trained on smaller dataset and our model is overfitting. We are not able to train Autoencoder(Neural Network) model on full dataset due to our system/memory/gpu constraints. If it would be trained on whole dataset od amazon product review, it would give good results on test dataset also.
As our Autoencoder model is not giving good results, so ensemble model is also not giving better accuracies as it takes results of Autoencoder model as inputs
For Inductive Matrix Factorisation model, as we have not taken good and sufficient features of users and items(feature selection) and haven't normalized it properly due to time contraints, it is not giving better result than normal matrix factorization model.

7 Things we learnt

We have read several recent research papers on recommender systems published in NIPS and RecSys tackling cold start, sparseness and handling millions of users. We have got an exposure to real research currently going on in this field. We have learned to implement a model in tensorflow using its various libraries which make our implementation simple and easy. We know now several databases to learn model and how to preprocess these. From the result and plots of this project, we learn to find optimal values of hyper-parameters and number of iterations to train the model (Elbow method).

8 Future Work

Cold start problem can be tackled more rigorously by considering the more number of relevant features of items and users(feature selection) and normalizing them appropriately in order to improve accuracy of recommendations and minimize the errors. So, Inductive matrix completion technique can be improved by applying techniques of feature selection and normalization. Similarly, Autoencoders can also be used to incorporate features of items and users [3] considering techniques of feature selection.

In order to improve accuracy of both models and combined ensemble model, whole learning algorithm can be made online, i.e. if we train the ensemble model (implicitly training both models as we train ensemble model) without considering results of both models as features to ensemble model.

References

1. Deep Learning for Recommender Systems by Alexandros Karatzoglou and Balzs Hidasi published in RecSys 17, August 2731, 2017, Como, Italy.

2. Ask the GRU: Multi-task Learning for Deep Text Recommendations - 2016 By Trapit Bansal, David Belanger, Andrew McCallum

3. AutoRec: Autoencoders Meet Collaborative Filtering

4. Training Deep AutoEncoders for Collaborative Filtering

5. Fast and Sample Efficient Inductive Matrix Completion via Multi-Phase Procrustes Flow

6. DropoutNet: Addressing Cold Start in Recommender Systems