Amazon Instant video Recommendation

First_Report

Divyaprakash Dhurandhar
dd839@scarletmail.rutgers.edu
Rutgers Univesity, New Brunswick

ABSTRACT

Recommendation systems can take advantage of social media in various ways. Recommendation systems are defined as the techniques used to predict the rating one individual will give to an item or social entity. These items can be books, movies, restaurants, and things on which individuals have different preferences. Prime Video, also referred to as Amazon Prime Video, is an Internet video on demand service that is developed, owned, and operated by Amazon. In this report, various models are presented like ExtraTreesRegressor, XGBRegreesor, Matrix Factorization, SVD, Quasi SVD and Probabilistic Matrix Factorization in predicting the rating and provide top video recommendations based on the predicted ratings. Using the review text, a BiLSTM model was implemented with attention mechanism to predict the rating. Finally, the ensemble of top 2 model ratings is taken to get the top recommendations. A recommendation system was built with better accuracy to predict the ratings, users would give for videos which they haven′t used before and to create a recommendation list for each user. A simple strategy to create such a recommendation list for a user is to predict the ratings on all the items that user didn′t buy before, then rank the items in descending order of the predicted rating, and finally take the top items as the recommendation list.

KEYWORDS

LSTM, Attention, ExtraTreesRegressor, XGBRegressor, SVD, Matrix Factorization, Probabilsitic Matrix Factorization, Video Recommendation

1 INTRODUCTION

Recommender systems are information filtering systems that deal with the problem of information overload by filtering vital information fragment out of a large amount of dynamically generated information according to user′s preferences, interest or observed behavior about an item. Recommender system has the ability to predict whether a particular user would prefer an item or not based on the user′s profile. Recommender systems are beneficial to both service providers and users. They reduce transaction costs of finding and selecting items in an online shopping environment. Recommendation systems have also proved to improve the decision making process and quality. In e-commerce setting, recommender systems enhance revenues, for the fact that they are effective means of selling more products. E—commerces, such as Amazon or eBay are putting a lot of money into recommender systems. They are building great teams just to focus on improving the accuracy of their recommenders, because by doing so, users are much more tempted to buy more things.

Recommender systems are widely used these days in e-commerce, for the purpose of targeted advertisement. Based on each user′s profile, previous purchase history, and online behaviors, they recommend products which they are likely to prefer. For example, Amazon.com recommends related products such as books, and Amazon Prime recommends videos that each user is interested in. There are many methods to recommend the video on Amazon prime. We have implemented various algorithms like ExtraTreesRegressor, XGBRegreesor, Matrix Factorization, SVD, Quasi SVD, Probabilistic Matrix Factorization and bagged all the results of models which gave better results to get a new good model.

For review-text, I have used 'added the attention' method to our biLSTM model which predict the ratings. Recurrent Neural Network (RNN) is one of the most popular architectures used in Natural Language Processing (NLP) tasks because its recurrent structure is very suitable to process variable length text. RNN can utilize distributed representations of words by first converting the tokens comprising each text into vectors, which form a matrix. This matrix includes two dimensions: the time-step dimension and the feature vector dimension.

2 RELATED WORK

Lot of research work has been done in recommendation systems mainly in the e-commerce field. Content Filtering creates a profile for each user or product to characterize its nature. Collaborative Filtering analyzes relationships between users and inter-dependencies among products to identify new user-item associations. Collaborative filtering is generally more accurate than content filtering however, it suffers from cold start problem. (If new user exist and does not have any inter-dependencies among others, we canfit recommend anything).

For rating prediction based on the review, a lot of research has been done in neural networks like using BiLSTM, dropout, embedded vectors, etc. Attention is one of the most influential ideas in the Deep Learning community. Even though this mechanism is now used in various problems like image captioning and others, it was initially designed in the context of Neural Machine Translation using Seq2Seq Models. The central idea behind Attention is not to throw away those intermediate encoder states but to utilize all

Conference'17, July 2017, Washington, DC, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

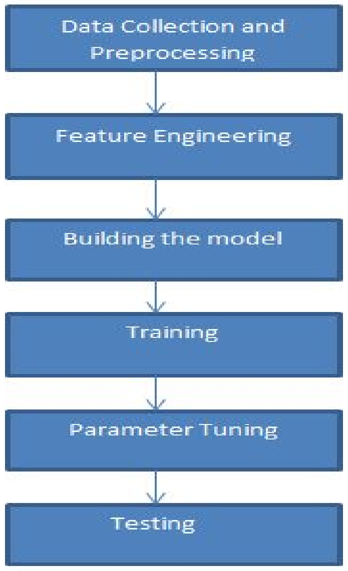
the states in order to construct the context vectors required by the decoder.
Text classification is one of the fundamental task in Natural Language Processing. The goal is to assign labels to text. It has broad applications including topic labeling (Wang and Manning, 2012), sentiment classification (Maas et al., 2011; Pang and Lee, 2008), and spam detection (Sahami et al., 1998). More recent approaches used deep learning, such as convolutional neural networks (Blunsom et al., 2014) and recurrent neural networks based on long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) to learn text representations.

3 PROBLEM FORMALIZATION

Mathematically formalize your research problem.

There are mainly 6 steps in solving the recommendation problem

Flow Diagram.



1. Data Collection and Preprocessing:
The dataset consists of 37126 reviews. There are Users and Videos. Ratings are on a scale of 1-5 and have been obtained from the Researchers from UC San Diego released a complete Amazon dataset that is publicly available online (http://jmcauley.ucsd.edu/data /amazon/). This dataset contains user reviews (numerical rating and textual comment) towards amazon products on 24 product categories, and there is an independent dataset for each product category. We have used the small subsets for experiment (the 5-core dataset) on the website, which can be downloaded directly from the website. Basically, each entry in a dataset is a user-item interaction record, including the following fields:

1. user-id : which is denoted as “reviewerID” in the dataset
2. product-id: which is denoted as “asin” in the dataset
3. rating: a 1-5 integer star rating, which is the rating that the user rated on the product, it is denoted as “overall” in the dataset
4. review: a piece of review text, which is the review content that the user commented about the product, it is denoted as “reviewText” in the dataset
5. title: the title of the review, which is denoted as “summary” in the dataset
6. timestamp: time that the user made the rating and review
7. helpfulness: contains two numbers, i.e., [users that think this review is not helpful, users that think this review is helpful]

Amazon Instant Video dataset was selected from the set of available datasets.
The data is split into train (70%) and test (30%).

2. Feature Engineering:
After some feature engineering, came up with new features such as count of each user, count of each video, time of review, time of review whether on weekend(Saturday or Sunday) or not. For Review Text, removed the html tags, punctuation marks, stop words and tokenized the words as part of feature engineering. Tried with Glove vector embedding but it is time-consuming.

3. Building the model:

1.ExtraTreesRegressor: A decision tree is usually trained by recursively splitting the data. Each split is chosen according to an information criterion which is maximized (or minimized) by one of the ′splitters′. These splitters usually take the form of x_i>t, where t is a threshold and x_i, indicates the i—th component of observation. Decision trees, being prone to overfit, have been transformed to random forests by training many trees over various sub-samples of the data (in terms of both observations and predictors used to train them). The main difference between random forests and extra trees (usually called extreme random forests) lies in the fact that, instead of computing the locally optimal feature/split combination (for the random forest), for each feature under consideration, a random value is selected for the split (for the extra trees). This leads to more diversified trees and less splitters to evaluate when training an extremely random forest. But training using extra trees regressor lead to overfitting on the training data.
2. XGBRegressor: XGBRegressor is a scalable and accurate implementation of gradient boosting machines and it has proven to push the limits of computing power for boosted trees algorithms as it was built and developed for the sole purpose of model performance and computational speed. This algorithm also got overfitted to training data but performance is better than ExtraTreesRegressor
3. Quasi-SVD: One way to handle the scalability and sparsity issue created by CF is to leverage a latent factor model to capture the similarity between users and items. Essentially, we want to turn the recommendation problem into an optimization problem. We can view it as how good we are in predicting the rating for items given a user. One common metric is the Root Mean Square Error (RMSE). The lower the RMSE, the better the performance. Since we do not know the rating for the unseen items, we will temporarily ignore them. MSE of traditional Collaborative Filtering model is quite high, we decided to use quasi-SVD approach instead. Here we use the cosine-distance to give the similarity between vectors. In this method, the output matrix is the dot product of three matrices.

Here 'U' is the user similarity matrix and 'P' is product similarity.

$$\Sigma_{m*n} = U_{m*m} \cdot \Sigma_{m*n} \cdot P_{n*n}$$

The advantage is that this method is quite interpretable. However, the drawbacks are also apparent: this algorithm is naive with a deficient performance. For a large data set, this algorithm needs m2+n2 times to calculate the similarity matrix.
4. SVD: In our recommendation system, there is such a matrix which has many scores from the users to the items. I hope to predict the targeted usersfi score to other unevaluated items and then recommend the items with the highest five scores. The advantage of SVD is that: users score matrix is a sparse matrix, so we can map the original data into a Low-dimensional space and then calculate the similarity of different items. This can help us reduce calculation complexity.

$$A_{m*n} = U_{m*m}\Sigma_{m*n}V^T_{n*n} \approx U_{m*k}\Sigma_{k*k}V^T_{k*n}$$

5. Matrix Factorization: Latent factor models are an alternative approach that tries to explain the ratings by characterizing both items and users on, say, 20 to 100 factors inferred from the rating patterns. For products, the discovered factors might measure obvious dimensions such as candy vs drinks, or adult food vs childrenfis; For users, each factor measures how much the user likes the product that scores high on the corresponding movie factor. Using the latent factor model, we transform the way to calculate the similarity of users and products. The features become more stable and condense.

We first create two new metrics, user—latent_factor and product—latent_factor. p, q are the total number of users and products, k is the number of latent factors. So, every element in the target matrix can be calculated as: And the target matrix is shown as. Next, we need to create the objective function based on the least-square method to minimize the loss:

$$e_{ij}^2 = \left(r_{ij} - \sum_{k=1}^{K} p_{ik}q_{kj}\right)^2 + \frac{\beta}{2}\sum_{k=1}^{K}\left(||P||^2 + ||Q||^2\right)$$

The system adjusts the model by fitting the previously observed ratings. However, the goal is to generalize those previous ratings in a way that predicts the unknown ratings. Thus, the system should avoid overfitting the observed data by regularizing the learned parameters by adding L2 term. The constant beta controls the extent of regularization and is usually determined by cross-validation. Next, we use stochastic gradient descent to optimize the objective function. Parameter beta is used to control the magnitudes of the user-feature and item-feature vectors such that P and Q would give a good approximation of R without having to contain large numbers. In practice, beta is set to some values in the range of 0.02. The new update rules for this squared error are as follows:

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

Where alpha is the stochastic learning rate and e is the error term. In the iteration, when the change in loss is larger than 0, the learning rate increases by 5%; if delta-loss is smaller than 0, it means the new loss is becoming larger. The learning rate decreases by 50% so that the loss can converge.
6. Probabilistic Matrix Factorization: Probabilistic Matrix Factorization is similar to Matrix Factorization. It is nothing more than a Matrix Factorization assuming the distribution of user and video are Gaussian. Rij represents the rating of user i for video j. U and V be latent user and video feature matrices, with column vectors Ui and Vj representing user-specific and video-specific latent feature vectors respectively. Iij is the indicator function that is equal to 1 if user i rated video j and equal to 0 otherwise.

When optimizing this function, the standard errors are fixed, so the objective function is:

$$\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij} - U_i^T V_j)^2 + \frac{\lambda_U}{2}\sum_{i=1}^{N}\| U_i \|^2 + \frac{\lambda_V}{2}\sum_{j=1}^{M}\| V_j \|^2$$

Gradient descent process is the same with basic MF. PMF does better than MF for sparse matrices. The assumption of Gaussian makes it more accurate to predict. But for our dataset, the chosen data is not sparse, so the performance of PMF is almost the same with MF.

For different sets of features, we trained different classifiers. The choice of the features is based on the validation set accuracy. Finally, a bagging model averaging top 2 models results is used which gave better results.

4. Training:
Divided the training data into train and test dataset with train_test_split ratio of 0.7

5. Parameter Tuning :
we have GridSearchCV with no of folds =5 and got the best parameters

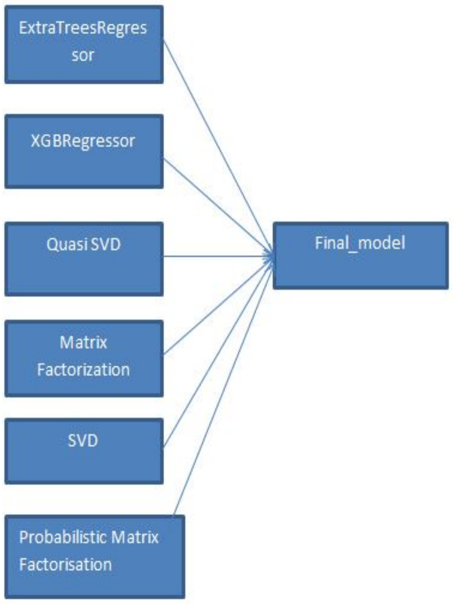
6. Testing:
Tested the data on the 0.3 percent of data and evaluated the results for various models.

4 THE PROPOSED MODEL

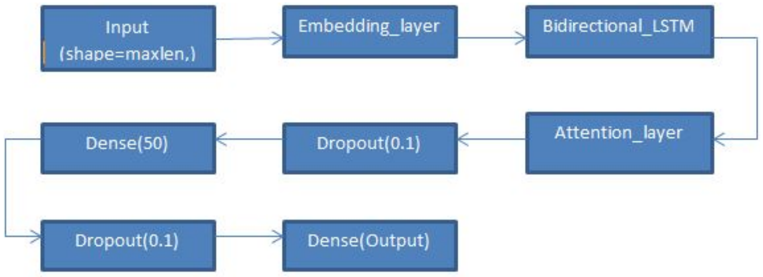
The proposed method in this paper take mainly two types of features 1. userid, videoid, time of rating, no of users, no of videos, review on weekend or not 2. Review Text(NLP Feature)

For the first kind of features: We used various models like ExtraTreesRegressor, XGBRegreesor, Matrix Factorization, SVD, Quasi SVD, Probabilistic Matrix Factorization to predict the user ratings. ExtraTreesRegressor and XGBRegressor got overfitted to the training data. So, we tried with SVD which gave better results. Then we implemented latent factor models. The algorithm will find the hidden factors that influence the userfis preference like the type of video, length of the video and so on for us. Here, no video description is required, only userfis history is good. Matrix Factorisation and Probabilistic Matrix Factorization performed better than all the methods. The average of both the ratings obtained by MF and PMF gave the best results among all the models. We sorted the results to get the top recommended videos. We took a subset of the features in every model and features are chosen based on validation results.

Flow Diagram.



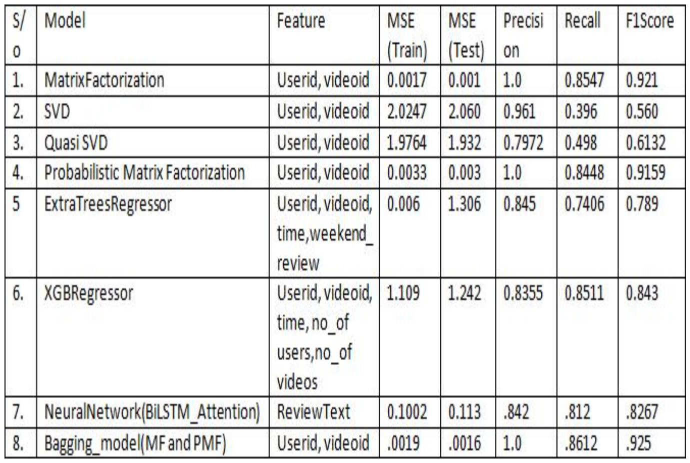
For the NLP Based feature, we first preprocessed the data, removed the stop words, html tags, punctuations etc. Tokenized the sentences to words and used BiLSTM with Attention mechanism to predict the ratings.



We have averaged the ratings using top two best models and recommended the top videos to the user.

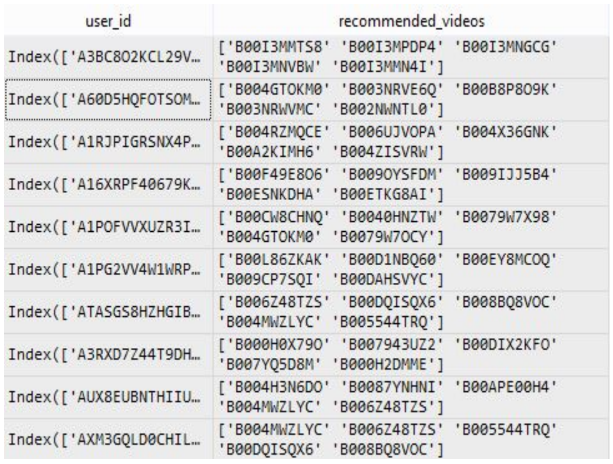
5 EXPERIMENTS

Results.



S/o	Model	Feature	MSE (Train)	MSE (Test)	Precision	Recall	F1Score
1.	MatrixFactorization	Userid, videoid	0.0017	0.001	1.0	0.8547	0.921
2.	SVD	Userid, videoid	2.0247	2.060	0.961	0.396	0.560
3.	Quasi SVD	Userid, videoid	1.9764	1.932	0.7972	0.498	0.6132
4.	Probabilistic Matrix Factorization	Userid, videoid	0.0033	0.003	1.0	0.8448	0.9159
5	ExtraTreesRegressor	Userid, videoid, time, weekend_review	0.006	1.306	0.845	0.7406	0.789
6.	XGBRegressor	Userid, videoid, time, no_of users, no_of videos	1.109	1.242	0.8355	0.8511	0.843
7.	NeuralNetwork(BiLSTM_Attention)	ReviewText	0.1002	0.113	.842	.812	.8267
8.	Bagging_model(MF and PMF)	Userid, videoid	.0019	.0016	1.0	.8612	.925

The top 5 recommended videos for the users will look like the following:



user_id	recommended_videos
Index(['A3BC8O2KCL29V...	['B00I3MMTS8' 'B00I3MPDP4' 'B00I3MNGCG' 'B00I3MNVBW' 'B00I3MMN4I']
Index(['A60D5HQFOTSOM...	['B004GTOKM0' 'B003NRVE6Q' 'B00B8P8O9K' 'B003NRWVMC' 'B002NWNTL0']
Index(['A1RJPIGRSNX4P...	['B004RZMQCE' 'B006UJVOPA' 'B004X36GNK' 'B00A2KIMH6' 'B004ZISVRW']
Index(['A16XRPF40679K...	['B00F49E8O6' 'B009OYSFDM' 'B009IJJ5B4' 'B00ESNKDHA' 'B00ETKG8AI']
Index(['A1POFVVXUZR3I...	['B00CW8CHNQ' 'B0040HNZTW' 'B0079W7X98' 'B004GTOKM0' 'B0079W7OCY']
Index(['A1PG2VV4W1WRP...	['B00L86ZKAK' 'B00D1NBQ60' 'B00EY8MCOQ' 'B009CP7SQI' 'B00DAHSVYC']
Index(['ATASGS8HZHGIB...	['B006Z48TZS' 'B00DQISQX6' 'B008BQ8VOC' 'B004MWZLYC' 'B005544TRQ']
Index(['A3RXD7Z44T9DH...	['B000H0X79O' 'B007943UZ2' 'B00DIX2KFO' 'B007YQ5D8M' 'B000H2DMME']
Index(['AUX8EUBNTHIIU...	['B004H3N6DO' 'B0087YNHNI' 'B00APE00H4' 'B004MWZLYC' 'B006Z48TZS']
Index(['AXM3GQLD0CHIL...	['B004MWZLYC' 'B006Z48TZS' 'B005544TRQ' 'B00DQISQX6' 'B008BQ8VOC']

6 CONCLUSIONS AND FUTURE WORK

ExtraTreesRegressor and XGBRegressor got overfitted to the training data. Matrix Factorisation and Probabilistic Matrix Factorization performed better than all the methods. The average of both the ratings obtained by MF and PMF gave the best results among all the models.
In this project, we have taken the average of the ratings with top two models, there can be more efficient ways to combine the results and get good results like stacking which can be done as future work. Also, this is done on offline data whereas in the real world, in the companies like Amazon, a huge amount of data will be accumulating data by data, so online streaming recommendation is useful, so this can be implemented as part of future work.

REFERENCES

[1] A. Paterek, Improving Regularized Singular Value Decomposition for Collaborative Filtering, Proc. KDD Cup and Workshop,

ACM Press, 2007, pp. 39-42 [2] Mukund Deshpande and George Karypis. 2004. Item based top n recommendation algorithms. TOIS (2004)
[3] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. Computer (2009).
[4] Dawen Liang, Jaan Altosaar, Laurent Charlin, and David M Blei. 2016. Factorization meets the item embedding: Regularizing matrix factorization with itemco-occurrence. In RecSys. 59–âĂŞ66.
[5] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In EMNLP. 1532—1543.
[6] https://papers.nips.cc/paper/7181-attention-is-all-you-need.pd
[7] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. 2008. Largescale parallel collaborative filtering for the netflix prize. In International Conference on Algorithmic Applications in Management. 337—348.