Recommendation System for E-commerce

CS:550 Massive Data Mining

Submitted by:

Jawahar Pinnelli (jp2141)

Viswajith Menon (vm623)

Sai Pradyumn Shrivastava (ss4369)

Objectives/Aim

- We experiment with some well-known recommendation models.
- Compare their performance.
- Analyse their different recommendation strategies.
- Hybridize their results to make contextual and relevant recommendations.

Data set description

- Data Source: https://nijianmo.github.io/amazon/index.html

Chosen Product - Cellphones and Accessories.

This dataset contains product reviews and metadata from Amazon, including 1.12 million reviews spanning May 1996 - July 2018.

This dataset includes reviews (ratings, text, helpfulness votes), product metadata (descriptions, category information, price, brand, and image features), and links (also viewed/also bought graphs).

- Number of Ratings – 1128437
- Number of products – 48186
- Number of Reviewers – 157212
- Average of ratings across all products – 4.221383205265336

Statistics	Rating
Range	0 - 5
Minimum	1.0
Maximum	5.0
Mean	4.221383205265336
Median	5.0
Variance	1.5176205932903861
Standard Deviation	1.2319174458097368

Dataset analysis

Recommendation Algorithms

- Singular Value Decomposition
- SVD++
- Co Clustering
- Self-Attentive Sequential Recommendatio

Stepwise overview of the Architecture.

- **Data Cleaning and Preprocessing**
- Training and Testing dataset
- Training and Evaluation
- Rating Prediction
- Item Prediction

- We read all the user, item, rating, review_time columns from the dataset.
- Then, convert that to a list of User, Item entries, sorted by when they reviewed it.

Stepwise overview of the Architecture.

- ○ Data Cleaning and Preprocessing
- ➢ **Training and Testing dataset**
- Training and Evaluation
- Rating Prediction
- Item Prediction

- Training dataset = 0.8 X Entire dataset
- Testing dataset = 0.2 X Entire dataset

Stepwise overview of the Architecture.

- ○ Data Cleaning and Preprocessing
- ○ Training and Testing dataset
- ➢ **Training and Evaluation**
- Rating Prediction
- Item Prediction

- Using the split train and test data, we train the models and evaluate their performance.
- We use the RMSE and MAE

Stepwise overview of the Architecture.

- ○ Data Cleaning and Preprocessing
- ○ Training and Testing dataset
- ○ Training and Evaluation
- ➢ **Rating Prediction**
- Item Prediction

Using several rating prediction models, we predict the ratings for the items in the training and testing dataset.

Stepwise overview of the Architecture.

- Data Cleaning and Preprocessing
- Training and Testing dataset
- Training and Evaluation
- Rating Prediction
- **Item Prediction**

For SVD, SVD++, and Co-Clustering item list is generated by sorting a list of products according to their predicted rating.

SASRec considers recent context and patterns in sequences to predict the next n most likely items to continue the sequence of recent purchases.

Put together, they could provide contextual and relevant items to users.

Rating Prediction Algorithms

SVD

- The SVD algorithm to factorizes the user-item matrix into three matrices, one representing the users, one representing the product and one that contains singular values that represent the importance of the latent factors in the user-item matrix.

SVD++

- The SVD++ algorithm, an extension of the SVD algorithm. The difference here is that it considers the implicit ratings.
- To incorporate implicit feedback data, the algorithm introduces an additional matrix Y that represents the implicit feedback data. Each row in the Y matrix represents a user's interactions with the implicit feedback sources. Here, an implicit rating describes the fact that a user 'u' rated an item 'j', regardless of the rating value.

Co-Clustering

- The co-clustering algorithm in the Surprise Python package tries to group similar users and similar items based on the pairwise interactions.
- Clusters are assigned using a straightforward optimization method, much like k-means.

Rating Prediction Algorithms

SVD

$$A = P\Sigma Q^T$$

P : User-to-concept similarity matrix
Q : Item-to-concept similarity matrix

$$\hat{r_{ui}} = \mu + b_i + b_u + (q_i)^T p_u$$

r^ui : predicted rating

μ : average rating of user *u*

b_i : bias in rating of item *i*

b_u : bias in rating of user *u*

p_u : user factors

q_i : item factors

A : rating matrix

SVD++

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \left(p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j \right)$$

r_{ui} : predicted rating

μ : average rating of user *u*

b_i : bias in rating of item *i*

b_u : bias in rating of user *u*

Y_j : The new set of item factors

that capture implicit ratings.

Co-Clustering

$$\hat{r}_{ui} = \overline{C_{ui}} + (\mu_u - \overline{C_u}) + (\mu_i - \overline{C_i}),$$

Where, C_{ui} is the average rating of co-cluster, C_u is the average rating of the user's cluster and C_i is the average rating of the item's cluster

Performance Metrics

- Rating Prediction Evaluation -

	RMSE	MAE
SVD	1.1453	0.8608
SVD++	1.1433	0.8454
CoClustering	1.2344	0.8545

- Item Recommendation Metrics -

➢ **Precision**: 0.8333097452272795

➢ **Recall**: 0.926464377854501l

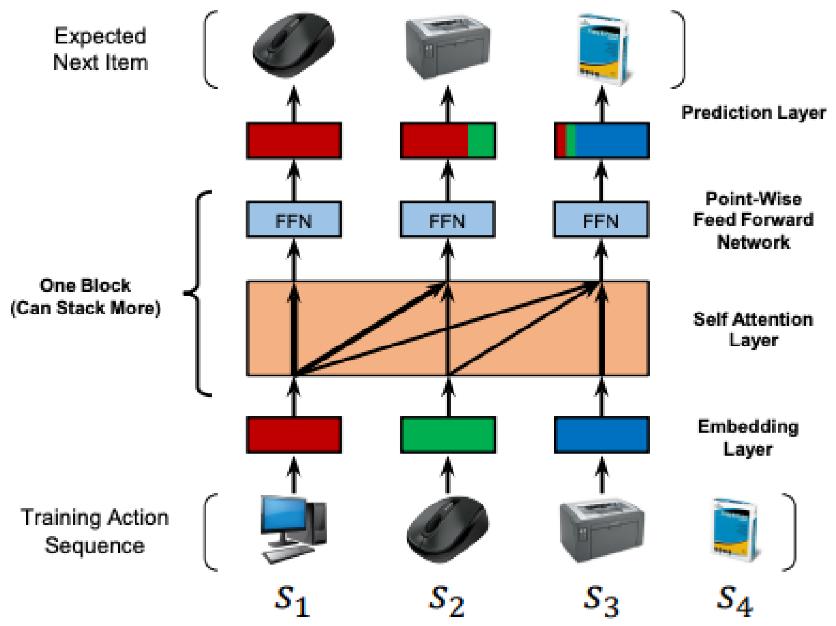
➢ **F-Score**: 0.8774214651140273

Predicted Ratings

	SVD	SVD++	Co-Clustering
User ID : A5AL9MYTWU5R9 ProductID :B015FLFC56	**True Value**: 5 **Predicted** : 4.50	**True Value**: 5 **Predicted** : 4.52	**True Value**: 5 **Predicted** : 5
User ID : AWRKVZQD2T9V5 ProductID :B0146G1M9Q	**True Value**: 5 **Predicted** : 4.50	**True Value**: 5 **Predicted** : 4.19	**True Value**: 5 **Predicted** : 3.43
User ID : A241FL95ZO6DQY ProductID :B00XIJRCOM	**True Value**: 4 **Predicted** : 4.04	**True Value**: 4 **Predicted** : 4.18	**True Value**: 4 **Predicted** : 3.59

Self Attention Sequential Recommendation

- SASRec is a self-attention based Transformer model for sequential recommendation.
- Using the attention mechanism, it can account for long-term dynamics by learning to pay more attention to a small subset of the input sequence.
- It can predict items that can best continue recent purchase context, while matching item relevant purchase patterns.



Self-Attentive Sequential Recommendation

- **Input:** the sequence of items with the last item withheld.
- **Output:** the sequence of items including the last item.

$$\mathcal{S}^u = (\mathcal{S}_1^u, \mathcal{S}_2^u, \ldots, \mathcal{S}_{|\mathcal{S}^u|}^u)$$

$$(\mathcal{S}_1^u, \mathcal{S}_2^u, \ldots, \mathcal{S}_{|\mathcal{S}^u|-1}^u)$$

$$(\mathcal{S}_2^u, \mathcal{S}_3^u, \ldots, \mathcal{S}_{|\mathcal{S}^u|}^u)$$

- Each user's data is transformed into a fixed length sequence, by truncating long ones, and padding short ones.

Self-Attentive Sequential Recommendation

Model Architecture:

- Each item in the sequence is represented with an embedding matrix $M \in R^{|I| \times d}$, and a positional embedding.
- Two attention layers are employed to learn item dependencies.
- A point-wise feed-forward network with ReLU transforms the input between layers.
- Dropout and layer normalization is done to avoid overfitting.

- NDCG@10: 0.356, Hit rate@10: 0.555

Conclusion:

- Transformer based sequential recommendation can model complex item relationships to recommend items that fit the current context.
- Rating prediction algorithms are useful in creating a list of most relevant items for a user.
- However, to make a list of items that are both relevant to the user's current context and is predicted to give a high rating, either algorithm cannot model for these two goals directly.
- By running rating prediction on the items recommended by SASRec, users could filter for items with high confidence, as suggested by an independent algorithm to the one making the item-list recommendation.

	SVD	SVD++	Co-Clustering
User ID : A5AL9MYTWU5R9 ProductID :B015FLFC56	True Value: 5 Predicted : 4.50	True Value: 5 Predicted : 4.52	True Value: 5 Predicted : 5
User ID : AWRKVZQD2T9V5 ProductID :B0146G1M9Q	True Value: 5 Predicted : 4.50	True Value: 5 Predicted : 4.19	True Value: 5 Predicted : 3.43
User ID : A241FL95ZO6DQY ProductID :B00XIJRCOM	True Value: 4 Predicted : 4.04	True Value: 4 Predicted : 4.18	True Value: 4 Predicted : 3.59

Thank You!

References:

- Kang Wang-Cheng, Julian McAuley. "*Self-Attentive Sequential Recommendation*" Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), IEEE, 2018, pp. 197-206, doi: 10.1109/ICDM.2018.00028.
- Simon Funk. 2006. Matrix Factorization. (2006). https://sifter.org/~simon/journal/20061211.html
- Python. 2009. Pandas. (2009). https://pypi.org/project/pandas/
- Amazon Dataset (Cellphones and Accessories): https://nijianmo.github.io/amazon/index.html