Movie Recommender System

Project Report

Introduction

- The objective of a Recommender System is to predict user preferences and recommend top items.
- Used in a wide variety of areas - movies, books, music, apparel, consumer products, mobile apps, games.

Benefits

- Increased revenue
- Improved customer engagement
- Personalization
- Customer retention
- Competitive advantage

Approaches

There are 2 widely used approaches to building recommendation systems.

- Content-based - Based on attributes of the items and user’s profile built with specific preferences.
- Collaborative Filtering - Based on user’s past behavior.

Collaborative Filtering

- **User-based**: measure the similarity between target users and other users
- **Item-based**: measure the similarity between the items that target users rates/ interacts with and other items

User-based CF

- Similarity between 2 users is calculated using Pearson correlation. Following is the formula.

$$u_{ik} = \frac{\sum_j (v_{ij} - v_i)(v_{kj} - v_k)}{\sqrt{\sum_j (v_{ij} - v_i)^2 \sum_j (v_{kj} - v_k)^2}}$$

Pearson Correlation (https://goo.gl/y93CsC)

- Recommendations are then made from the movies highly rated by the most similar user that haven't yet been watched by the active user.

Singular Value Decomposition

- Uses Matrix Factorization to extract latent features. This results in dimensionality reduction
- $$\underset{m \times n}{\overset{\hat{X}}{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}}} \approx \underset{m \times r}{\overset{U}{\begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \\ u_{m1} & & u_{mr} \end{pmatrix}}} \underset{r \times r}{\overset{S}{\begin{pmatrix} s_{11} & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & s_{rr} \end{pmatrix}}} \underset{r \times n}{\overset{V^{\top}}{\begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \\ v_{r1} & & v_{rn} \end{pmatrix}}}$$

Singular Matrix Decomposition(http://www.cs.carleton.edu/cs_comps/0607/recommend/recommender/images/svd2.png)

- X denotes the utility matrix,
- U is a left singular matrix, representing the relationship between users and **latent factors.**
- S is a diagonal matrix describing the strength of each latent factor
- V transpose is a right singular matrix, indicating the similarity between items and latent factors.

Evaluation Metrics - Accuracy

- RMSE - Root Mean Squared Error

The square root of the average of squared differences between prediction and actual observation.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

- MAE - Mean Absolute Error

The average over the test sample of the absolute differences between prediction and actual observation

$$\text{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

Evaluation Metrics - Top N

- While accuracy metrics like RMSE are widely used, they are poor estimators of online performance.
- Top N evaluation metrics are more useful in the real world.
- The top N items are presented to the user and evaluation is done based on subsequent user interaction with any of the top N items.

Evaluation Metrics - Top N

- Hit Rate - Number of successful hits from Top N / Number of users
- Average Reciprocal Hit Rate - Cumulative value of 1/rank / Number of users
- Cumulative Hit Rate - How often we recommended a movie the user actually liked (ignore low ratings)
- Hit Rate by Rating value - Break down of hit rate by rating value

Top N metrics selection

- As always, which metrics are the most relevant depends on the requirements of the system and design of the user interface that displays the recommendations.
- As an example, if the top N recommendations are laid out side by side and are treated equally, the hit rate might be enough.
- If the user has to scroll through recommendations, rankings of individual items need to be taken into account. The average reciprocal hit rate might be a better metric in this case.

Project metrics - Accuracy

The metrics were calculated using scikit surprise library methods.

- Direct accuracy metrics for RMSE and MAE

```
RMSE:  0.9033701087151802
MAE:  0.6977882196132263
```

- 5 fold cross validation results for RMSE and MAE

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	0.8953	0.8950	0.8974	0.8953	0.9027	0.8972	0.0029
MAE (testset)	0.6899	0.6892	0.6919	0.6900	0.6927	0.6907	0.0013

Project metrics - Top N

1. To compute the Top N metrics, scikit surprise's LeaveOneOut model selector was used. This holds out a specified number of observations (1 rating in our case).

2. Our model is trained without the left out ratings.

3. Top N recommendations are predicted by the model.

4. Metrics calculated by checking if the top N movies include the left out one in the first step.

Project metrics - Top N

- Following were the observed Top N metrics

```
Hit Rate:  0.029806259314456036

ARHR (Average Reciprocal Hit Rank):  0.0111560570576964

cHR (Cumulative Hit Rate, rating >= 4):  0.04960835509138381

rHR (Hit Rate by Rating value):
3.5 0.017241379310344827
4.0 0.0425531914893617
4.5 0.020833333333333332
5.0 0.06802721088435375
```

Top N metrics interpretation

- The top N metrics above appear to be on the lower side due to the sparsity of user ratings data.
- Best to combine metrics for more insight. In our case, the hit rate is 3% but the cumulative hit rate (cHR) is 5%. This is good as the cHR discards low ranked left out ratings.
- Which ones matter depend on the design of UI that displays top N items. ARHR rewards hits for being on top of the list.
- These metrics are the most meaningful when comparing different recommender implementations.

A/B Testing

- A recommender system with great accuracy and top N metrics isn’t still guaranteed to be successful if rolled out.
- Online A/B tests are extremely important to evaluate real world impact and tune the recommender system.
- Good A/B tests help companies to invest time and resources only on systems that actually work in the real world.
- A/B tests typically come up with estimates for CTR (Click Through Rate), CR (Conversion rate) and ROI (Return on Investment).

Summary

Explored / learned the following during the course of my Capstone project

- Multiple approaches to coming up with recommendations - User based CF and SVD.

- Compute Pearson correlation among users to determine similarity for User based Collaborative Filtering.

- The importance of ranking based metrics for recommender systems in the real world in addition to accuracy metrics.

- Interpret the metrics for recommender systems and pick the relevant metrics for a given use case.

- Accuracy and top N metrics can be used to narrow down the selections to a single implementation or a few. Online A/B tests are the actual indicators of the success of a recommender system.

Next steps

I would like to explore the following further

- Ways to transform the user ratings matrix to reduce the effects of sparsity.

- Deep Learning based approaches to building recommender systems.

Links & references

- My github repo with capstone project resources and other springboard submissions. https://github.com/lakshv77/springboard/

- Helpful post on overview and comparison of different approaches to implement recommendation systems. https://towardsdatascience.com/various-implementations-of-collaborative-filtering-100385c6dfe0

- Top N metrics evaluation was based on the material from this online course. https://www.udemy.com/building-recommender-systems-with-machine-learning-and-ai/