Recommendation Systems: Predicting Peer Movie Reviews

GROUP 6

11/24/2019

BAX 401 Introduction to Business Analytics

Zunfeng Huang
Mitesh Ranmal Jain
John Elmer Loretizo
Mayank Mani
Shalini Mishra
Prakriti Rastogi

Executive Summary

On October 2, 2006, Netflix started the Netflix Prize Competition which gave prominence to recommendation systems that are widely used by almost all tech and e-commerce giants today. These systems generate a list of recommended products that users will most likely consume or purchase. The algorithm works in 2 different ways – collaborative filtering, that uses community preferences for recommendations, and content-based filtering, that matches user characteristics to product attributes. With the help of the models created, we were able to establish the following conclusions:

- Recommendation engines rely on historical data. Hence, no recommendations can be made for new users or new products in the system – a classic cold start problem that recommendation engines face.
- As a rule of thumb, new users can be directed to top-rated or trending movies until such time that the system has enough data to provide them with a set of customized recommended items.
- New users can be asked to rate a few products from the catalog at the time of acquisition.
- A set of trusted sources should be established for accessing new users' data or ratings of new products.
- Collect and utilize content-based information to include new products in recommendations.
- Recommend the new product to a random set of customers and track preferences to identify the group that responds best.

Several techniques can be used to predict user behavior, each more sophisticated than the previous one. The primary challenge here isn't developing a recommendation engine, but instead, implementing them in an environment that enables these algorithms to perform at their best capacity.

Introduction

In 1996, Bill Gates wrote “content is king” in one of his essays describing the future of the internet as a marketplace. In this day and age, this phrase is especially relevant for all companies whose business model is centered on delivering personalized service, product, or experience. To compete and survive in the market, tech giants like Google, Facebook, Amazon among others use recommendation systems. These algorithms aim at suggesting the best products to the users in an effort to drive more profit. In this report, we will take a look at how existing ratings are used to predict user ratings for movies one has never seen before, particularly through the use of collaborative filtering techniques.

Problem Formulation

A recommendation engine filters the data based on different algorithms and then recommends the most relevant items to users. In the case of movie ratings, recommendation systems are typically used to map the users in a segment by finding similarities in the ratings given by them in the following steps:

- *The similarities can be calculated by using any of the methods - Cosine, Euclidian[1], Manhattan[2] distances or correlation between the users.*
- *These methods are augmented by using the original data, mean-centered or z-score data for the calculations.*
- *The similarity scores thus calculated are then used for predicting the ratings by applying either of the collaborative filters: user-user or item-item.*

[1] "ordinary" straight-line distance between two points.
[2] The distance between two points measured along axes at right angles

We applied the cosine similarity index for predicting the similarity scores and tested it against each of the original data, mean-centered or z-score data to predict which one was the closest to the ratings available. Specifically, we aim to demonstrate the following in this report:

- Predict group members' ratings for any 3 movies as well as predict movies for Winter's Bone, Son of Saul, and A Serious Man.
- Predict movie ratings for new users, in this case, Amy, Camille, and Shachi as well as predict their ratings after receiving new data.

Data Description

We have been provided with a list of movie ratings of all the students in our MSBA 2020 batch for selected 50 Academy Award Nominated Movies. The ratings are ordinal in nature, ranging from 1 to 5, with 1 being the lowest and 5 being the highest rating. Because the movies have been rated by a diverse group of people from different countries, the ratings do not share the same meaning and standards. Consequently, even without the demographic factor, ratings are highly subjective as preferences of one user may vary from that of the other.

Model Results

Data for all movies were leveraged in computing cosine similarity index for predicting the reviews of the three movies by each team member. Three collaborative filtering approaches - original data, z score and mean-centered were applied and assessed based on their respective RMSE[3]. We also validated the algorithms against the DBMI data. Based on within validation, the mean-centered cosine similarity index yielded the best results across users. The ratings for the group are as follows

Movie	Rastogi	Mishra	Mani	Huang	Jain

[3] RMSE – Root Mean Square Error

	Actual	*Predicted*	*Actual*	*Predicted*	*Actual*	*Predicted*	*Actual*	*Predicted*	*Actual*	*Predicted*
Inception	4	4.48	3	2.66	5	4.79	3	3.77	4	3.89
Avatar	3	3.81	3	2.7	4	4.16	5	4.8	3	3.26
The Wolf of Wallstreet	5	4.79	2	2.33	5	4.86	5	4.65	5	4.38

Mayank was found to be the kindest in his reviews whereas Shalini was harshest.

Classic recommender systems like collaborative filtering rely on historical data on each user or item in order to infer ratings of similar users/items even if those ratings are unavailable. On the arrival of new-users or new items, in this case predicting movie ratings for Winter's Soul, Son of Saul, and A Serious Man or for Amy, Shachi, and Camille joined, it suffers the cold start problem[4] (Wei et. al., 2016). We face two situations - item cold start for new movies or movies without any ratings from all existing users and user cold start for new users due to lack of historical data.

Contrary to getting inconclusive results when Amy, Camille, and Shachi just joined (Cosine similarity as 0), on incorporating given ratings for the new users in Question 4, we were able to obtain a non-zero similarity matrix making our results consumable.

Movie	Camill		Shachi		Amy	
	Predicted	*Rounded*	*Predicted*	*Rounded*	*Predicted*	*Rounded*
Inception	2.65	3	4.28	4	3.09	3
Avatar	2.71	3	4.14	4	3.02	3
The Wolf of Wallstreet	2.7	3	4.39	4	3.1	3

Challenges

Recommendation engines are not free of developmental and implementation issues:

[4] The problem that models or algorithms encounter when there are no initial data for a specific item of interest.

1. Cold Start is a very well-known problem being faced by companies trying to incorporate recommendations into their business model. Unknown preferences of new users and unknown ratings of new products can lead to less accurate predictions.
2. For most analysts, building a recommendation engine model is not a challenge but implementing it is a bigger issue for organizations. Infrastructural requirements of these models are high because they are computationally heavy, and this only increases with the increase in customer base.
3. Models aren't protected from shilling attacks from malicious users or competitors that can enter the system and give false ratings which in turn would mess with the predictions.
4. Recommendation engines do not work very well with sparse matrices, which is a common occurrence in companies with a huge customer base and an equally large product catalog. An apt example of this is e-commerce giant Amazon's data. This creates a 2-fold problem - not all customers buy all these products and even the ones who do buy tend to not go back to the portal for providing ratings.

Recommendations

1. Tackling new customers:
    a. *Recommend top-rated, trending, popular or most profitable movies*. Movie recommendations for new users can include categories for trending, popular or top-rated movies according to their demographics or location or both.
    b. *Initialize ratings during the acquisition process*. Ask customers for their ratings at the time of acquisition.
    c. *"Merge" ratings from other available sources*. Access new users' ratings from other trusted resources with their consent.

d. *Generalize customer base and recommendations*. Identify the characteristics of the most active customer base and translate their recommendations to new users.

2. Tackling new movies:
    a. *Incorporate content-based information*. By using tags for a movie (genres, actors, director, franchise, etc.), the new movies could be assigned with more generalized ratings before entering the system.
    b. *Promote new products*. The use of CRM or other marketing strategies to promote new products would increase engagement and data for the new product.
    c. *A/B testing for new products*. Recommend the new product to a random set of people and track their preferences to identify the set of likely customers.
3. Additional data collection:
    a. *Demographic data for customers*. Demographics and location data of customers can be used to recommend movies for new users with similar demographics.
    b. *Revenue per user*. Tracking users' spending for each movie (movie tickets, merchandise, etc.) would help in suggesting the highest revenue-generating movies to new users.
    c. *Content-based tags for movies*. Collect content-related information for all movies and use that to create tags for identifying similar categories of movies.

Conclusion

Recommendation engines are the new trendy kid on the block and every company wants to get in on the action in the hopes of increasing their revenue and customer retention, especially in the current scenario where every customer is looking for personalization. But just like every other fancy tech, REs have certain limitations and solutions. If the organization is committed to providing standard customer service, these hurdles can be overcome with infrastructure and dedication.

References

Guo, G., Zhang, J. and Thalmann, D. (2019). Merging trust in collaborative filtering to alleviate data sparsity and cold start.

Ji, K. and Shen, H. (2019). Addressing cold-start: Scalable recommendation with tags and keywords.

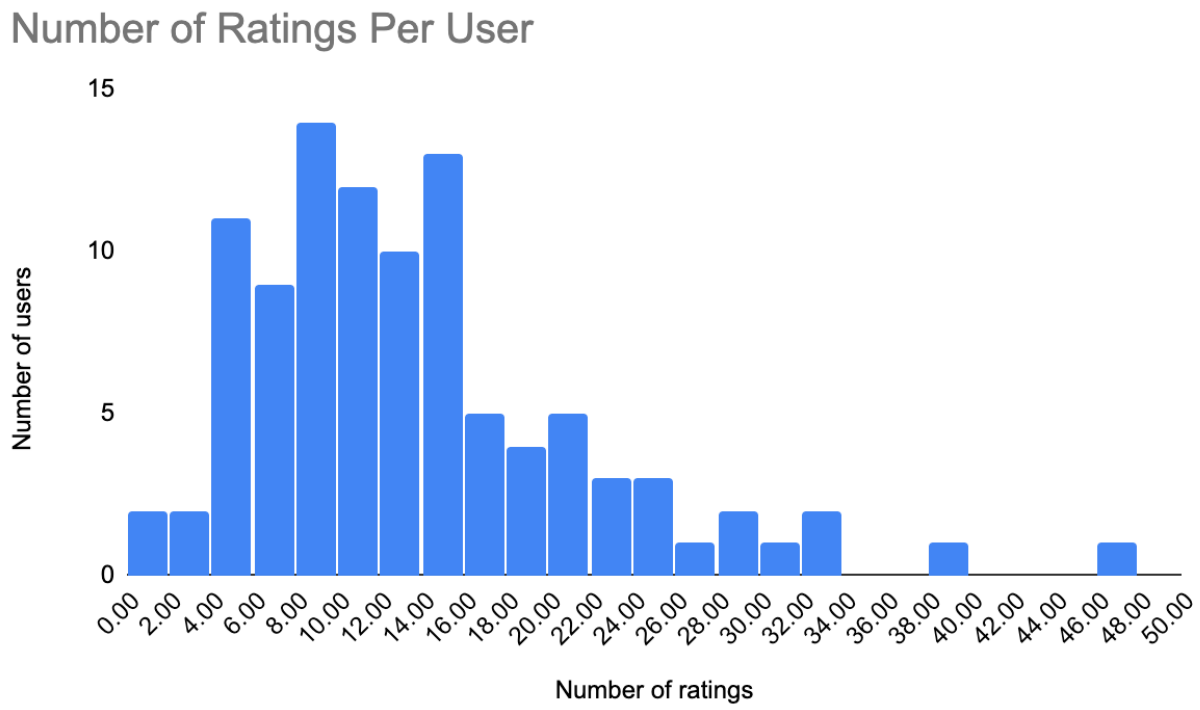
Khusro, Shah & Ali, Zafar & Ullah, Irfan. (2016). Recommender Systems: Issues, Challenges, and Research Opportunities.

Töscher, A., Jahrer, M. (2009). The BigChaos Solution to the Netflix Grand Prize

Wei, Jian & He, Jianhua & Chen, Kai & Zhou, Yi & Tang, Zuoyin. (2016). Collaborative Filtering and Deep Learning Based Recommendation System For Cold Start Items. Expert Systems with Applications.

Appendix

Rating distribution:
Mean number of ratings per user: 13.5
Mean rating: 4



Cosine similarity formula:

$$Cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \, \|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2} \sqrt{\sum_1^n b_i^2}}$$

where, $\vec{a} \cdot \vec{b} = \sum_1^n a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$ is the dot product of the two vectors.

It takes into account the angle between two vectors, which are ratings given by users 1 & 2 in this case and gives out a value between -1 and 1. The values of a & b represent the ratings (original, mean centered or z-scored) for each movie.

Formula: Cosine similarity scores

The predicted ratings are then found by using the similarity scores created above and are compared for different models:

$$P_{u,i} = \frac{\sum_v (r_{v,i} * s_{u,v})}{\sum_v s_{u,v}}$$

Formula: Prediction values using user-user collaborative filter

RMSE for each Group Member on the Three Variants of Cosine Similarity Index (Actual vs Predictions)

$$\boldsymbol{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\boldsymbol{P_i} - \boldsymbol{O_i})^2}{\boldsymbol{n}}}$$

Methodology	Rastogi	Jain	Mishra	Mani	Huang
Cosine Similarity Index	2.6345	1.7379	1.6499	2.2031	2.7440
Mean Centered Cosine Similarity Index	0.4283	0.3909	0.4659	0.4822	0.4752
Z Scored Cosine Similarity Index	0.4714	0.4136	0.4828	0.5308	0.5045

RMSE for DBMI Data Using Mean Centered Cosine Similarity Index

$$\boldsymbol{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\boldsymbol{P_i} - \boldsymbol{O_i})^2}{\boldsymbol{n}}}$$

RMSE = 0.5674