IIC 3633 - Sistemas Recomendadores
Paper Proyecto
On using association ruled-based recommendation with implicit-feedback per user

Maximiliano Friedl

Abstract

In this paper, we explore into using association rules for providing recommendations to users, based on implicit data gathered from an e-commerce environment. For a baseline benchmark, the performance was compared with different algorithms, such as most popular, ALS and matrix factorization.

Two different approaches to recommendation using association rules were tested. The first one did not pan out any good results. In the process of the second one, computational problems were encountered, making it unviable in the time span at hand.

1 INTRODUCTION

In e-commerce, product recommendation is something that has been vital for a long time. It seeks to make the traverse of all the items easier on the user, showing products that he might be interested in, resulting in a potential purchase. There are many approaches to this problem, and in this paper, we are going to dive into the problem using association rules.

Association rules are more fit to create recommendations based on items, rather than users, because of their nature. More specifically, association rules are perfect for generating recommendations based on a single item. For example, in the specific view of a single item, under the you might be interested in title.

Making this kind of recommendation requires some state. In this case, requires you to be on a single page of the e-commerce website. That is not the kind of data that we are working with, and thus, it was decided to explore into a different area, of making recommendations based on the user and their purchase history, instead of a current state, instead of the more obvious one.

The data could've been used to simulate the states mentioned beforehand, but it would result in something too boring and unoriginal, because, as it was mentioned, association rules thrill on single product based recommendations, because of their nature.

2 ASSOCIATION RULES

Association rule learning is a method to find relations between items in a dataset. It is based on the idea of transactions, and how often different items are present in the same transactions, together. They have the form of $\{X\} \rightarrow \{Y\}$, where the lefthand side is the base group of items, and the righthand side is the implication of the lefthand side, and also, another group of items.

Lets use an example: $\{Diapers\} \rightarrow \{Beer\}$. In this rule, we have two 1-item groups. The meaning of this is that the items in the right side are implied because of the items in the left side, and it will likely appear in a transaction, if the one on the left is present too. In particular, it is saying that if someone is buying diapers, it is likely to be buying beer too.

This rules area not crafted to be used in recommendations systems. Actually, their typical purpose is to aid in promotional pricing, or product placement, in traditional stores.

2.1 ASSOCIATION RULES METRICS

In order to differentiate between all the rules, and to generate them to begin with, there are multiple metrics. The following three are the most common, and the ones that will be used in this paper:

- **Support:** Represents the percentage of transactions in which the item or itemset is present. It is calculated as:

$$support(A) = \frac{transactionsWithA}{AllTransactions}$$

.

- **Confidence:** Given two items, confidence measures the percentage of times that item B was purchased, given that the item A was purchased. It is calculated as:

$$confidence(A \to B) = \frac{support(A, B)}{support(A)}$$

- **Lift:** Given two items, A and B, lift indicates whether there is a relationship between A and B, or whether the two items occur together in the same order simply by chance. Lift > 1 implies that there is a positive relationship between A and B. It can be calculated as:

$$lift(A, B) = \frac{support(A, B)}{support(A) * support(B)}$$

3 ASSOCIATION RULES RECOMMENDATION

Given the former defition for association rules, the two models used to generate recommendations based off of them are as follows.

3.1 FIRST APPROACH

In the first approach[1], the association rules were calculated based off the entire training dataset at once. Several calculations were performed, with varying ammounts for the value of minimum support and minimum confidence, and then they were sorted by higher values first.

Then, the rules inside each of this sets with rules were sorted by their respective lift. With this, a final set of rules, with varying support, confidence and lift is attained.

After this, for each user the items the user has bought are taken from the training dataset, and thrown into a list. For each rule set, we can check if the items bought by the user are present in any of the rules. Because of the ordering of the rules done before, it will check first on the highest lift rules, from the highest confidence and support sets.

If there is a match between an item bought by a user, and the items present in the lefthand side of a rule, we then recommend the righthand side of said rule. Then, the process is repeated until there are enough items in the recommendation list for the user.

3.2 SECOND APPROACH

In the second approach[2] the association rules were calculated for each user, based solely on the transactions made by this user in the past. A single association rule set is calculated per user, with a single minimum confidence and support.

After the rulesets are calculated for each user, this specific rulesets are used in the same method than the previous approach to generate recommendations for the users. But, this time, we take into account how many times each user has bought each item, giving priority to these that have been bought more.

3.3 COMPARISON BETWEEN METHODS

Despite the fact that both methods are pretty similar, they end up being a lot different. In the first method, we are not distinguishing between users at the moment of generating the rulesets, so it is similar to a collaborative filtering algorithm based on items. It has de disadvantage that it doesn't really learn particular behaviour or likings of the users, it just creates association between items, and then a user showed interest in an item, it then recommends the associations.

In the second method the system is learning specifically from the purchase history of a single user, and thus, it is able to generate associations that are based on the behaviour of said user. The down side is that the size of the dataset from which rulesets are created is much smaller, and a lot of transactions per user are required to achieve good rules. In the other hand, as it is learning exclusively from a single user's history, it has no room for novel recommendations, and the system is limited to recommending items that said user has bought in the past.

4 EXPERIMENTS

In order to determine if any of the approaches works in the context, several experiments were performed to measure the models. The metric of choice to determine the performance was mean average precision, or MAP.

4.1 DATASET

A dataset of 28544 transactions from an e-commerce website will be used for the experiments. An interesting part of the dataset is, it was collected 100% during the SARS CoV-2 pandemic, and thus, it may represent certain behaviours in the users that are not typical.

The dataset was split into a 75/25 train and test datasets to perform the experiments. The transactions are split among 361 different users, and 3045 unique

[1] based on: https://medium.com/datadriveninvestor/recommendation-system-using-association-rule-mining-for-implicit-data-6fba0f6c5012

[2] based on: https://medium.com/@jwu2/content-based-recommender-systems-and-association-rules-599843cb2fd9

Table 1 Summary table of the dataset.

Statistic	Valor
Number of users	361
Number of unique items	3045
Total transactions	28544
Mean transactions per user	79
Mean users per transaction	9
Sparsity	0.014%

items. In table 1, a summary of some statistics of the dataset is present.

4.2 MOST-POPULAR

As a baseline of performance, a most-popular approach was taken. Most-popular algorithms are traditionally implemented as an explicit feedback system, and thus, the dataset at hand was not ideal for them. Because of this, two metrics were crafted from the dataset, in order to generate a new column simulating a rating given by the user to each of the different items in it's buying history.

In the first one, the ammount of times a user has bought each item was taken as the metric. Then, the range of values for this column was mapped to a range of $[1, 5]$ to simulate a rating from 1 to 5 stars for each item from each user. An item that was bought more frequent, indicates that the user liked it more, and thus will have a higher rating.

The problem with the former method is, it is biased towards smaller, cheaper items. Let's take as an example two items that were present in the dataset, yoghurt and a 20kg bag of flour. One is an item that is frequently bought, and in big ammount, besides being super cheap. In the other hand, we have an item that last really long and is a lot more expensive than the first one. In order to try and compensate for the bias, the ammount of times an item was bought was multiplied by the price of each item, given more expensive items more of a chance to compete with the cheaper ones.

It is still not perfect, but given the columns present in the data,a it was the best that could be done to simulate a rating. After this multiplication was performed, the same map to a range of $[1, 5]$ was performed and used as a rating for each item by each user.

4.3 OTHER ALGORITHMS

In order to have a good mix of different algorithms to compare the models to, models for Alternating least squares, bayesian personalized ranking and logistic matrix factorization were trained and evaluated in the same dataset. The implementations used were from the

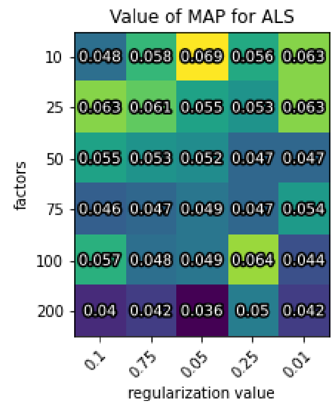

Figure 1. MAP values for different training hyperparameters of ALS.

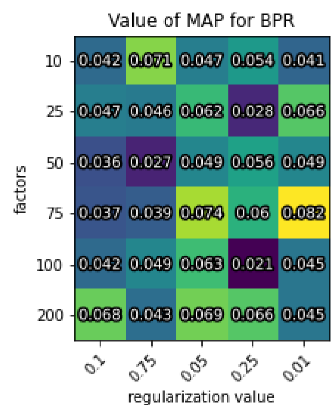

Figure 2. MAP values for different training hyperparameters of BPR.

`implicit`[3] library, in Python.

5 RESULTS

After training the different models and calculating it's respective MAP, the following results were found. As a baseline, the most-popular approach resulted in MAP values of 0.040 and 0.039 for the value and ammount approaches to simulating ratings, respectively.

As for ALS, BPR and LMF, several number of factors and regularization values were tested. The best results were 0.06, 0.082 and 0.062, respectively. A more in-depth and more detailed displaying of the MAP values for each one can be found in figures 1 through 3, with the values for ALS, BPR and LMF, respectively.

As for the association rule-based recommendations, for our first approach the results were not as expected, at all. The MAP values were so low, that they were

[3] https://implicit.readthedocs.io/en/latest/

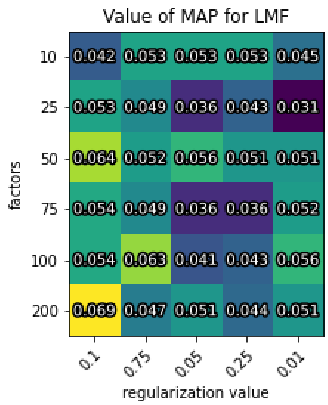

Figure 3. MAP values for different training hyperparameters of LMF.

rounded to 0.0. Despite the problems listed before with the method, it was not expected to have such atrocious results.

In the case of the second method, we were unable to actually perform the tests. Computational limitations were confronted. In an attempt to calculate the association rules for every user, we ran into RAM limitations. An alternate approach was to calculate each one, and then store it into disk for further use, but this approach was way too slow, as disk readings are just too slow compared to RAM readings.

Furthermore, the possibility of implementing the system as a lazy recommender was present. For this, the association rules of each user would have to be calculated on the fly, and based on that generate the recommendations at the same time. For each recommendation, we would have to go through the dataset and generate the rulesets, which is highly unpractical and slow.

6 FUTURE WORK

As it was mentioned, because of time restraints and computational power, it was not possible to get a real test of the second method for association-rule-based recommendation. And thus, it would be interesting to explore further into this idea, refine the concepts and keep working on it, as it probably has potential. It would be interesting to test this type of recommendation in a more natural and fitting space for recommendation rules, as item-based recommendation would be.

On another topic, since the beginning of this endeavour, the objective was to perform the recommendations using a deep learning approach. Because the ammount of data present did not allow for this method, it was then decided to explore into the association rule based recommendations. Because of this, as time goes on and more data is gathered, using a deep learning method on this data is desired.

7 CONCLUSION

Despite the fact that the results were not as expected, it still provided some insight into the working of association rules and how to perform recommendations with them. Besides that, it sets a ground-work for future explorations into the same topic.

On another topic, we could see that the highest MAP values were for BPR, but for the other algorithms, ALS and LMF, the values were not that far off of most-popular. This tells us two things, one, the simulation of a rating for each item performed was not bad. Second, despite being incredibly easy, the most-popular approach still yields good results.

8 REFERENCES

1. Takács, Gábor Tikk, Domonkos. (2012). Alternating least squares for personalized ranking. 10.1145/2365952.2365972.
2. Steffen Rendle, Christoph Freudenthaler, Zeno Gantner and Lars Schmidt-Thieme (2009). BPR: Bayesian Personalized Ranking from Implicit Feedback
3. Christopher C. Johnson. (2014). Logistic Matrix Factorization for Implicit Feedback Data