УДК 004.6
МРНТИ 28.23.17

PERSONALIZED TRAINING RECOMMENDATION SYSTEM BASED ON COLLABORATIVE FILTERING

TOREMURATULY A.[1], ZHAILKHAN M.M.[1], URPEKOVA A.Z.[2]

[1]Kazakh-British Technical University
[2]International University of Information Technologies

Abstract: *In this article we have covered many approaches of implementing personalized training recommendation system based on collaborative filtering. These techniques are consist of memory based methods, where we apply our statistical methods to the entire dataset to make predictions.*

We have considered such algorithms as cosine similarity and Pearson correlation. For cosine similarity we consider users data as vector of some collaborations in N dimensional space, where N is number of items. Then we calculate similarity of any two users as cosine of an angle between their vectors. This technique end up with good results, but anyway there is a problem, because of the matrix sparsity (empty interactions). Considering them as 0, impacts results even if we remove mean from each existing collaboration. Therefore, we also considered Pearson correlation which operates better with empty spaces in our data matrix. Here we try to find positive or negative trends between users and get correlation coefficient to predict rating.

At the end of article we have compared all techniques based on such approaches as measuring RMSE and MAE

Key words: *recommendation systems, collaborative filtering, cosine similarity, Pearson correlation, gym, trainings, sport*

СИСТЕМА ПЕРСОНАЛЬНЫХ РЕКОМЕНДАЦИЙ ТРЕНИРОВОК, ОСНОВАННАЯ НА КОЛЛАБОРАТИВНОЙ ФИЛЬТРАЦИИ

Аннотация: *В данной статье выборочно рассматриваются несколько подходов реализации персональной системы рекомендаций тренировок с помощью коллаборативной фильтрации. Главным направлением была выбрана фильтрация, основанная на работе с памятью, где авторы напрямую работают с данными и пытаются вычленить нужные связи, производя статистические методы на всем датасете целиком.*

В список рассмотренных алгоритмов входят такие как метод косинусного сходства и метод корреляции Пирсона. В первом случае мы рассматриваем каждого пользователя как N-мерный вектор, где N – это количество рассматриваемых тренировок. Далее мы считаем сходство между двумя конкретными пользователями как косинус угла между двумя их векторами. Данный подход дал достаточно хорошие результаты, однако пустые клетки разряженной матрицы сильно повлияли на результат, так как этот метод плохо работает. В случае же Корреляции Пирсона пытаемся найти позитивные или негативные тренды между юзерами и считаем коэффициент корреляции, который далее будет использован при прогнозе значений для пустых ячеек.

Конечный результат статьи – сравнить все вышеперечисленные методы и рассказать о плюсах и минусах каждого. Сравнения произведены с помощью подсчета таких метрик как RMSE и MAE.

Ключевые слова: *системы рекомендации, коллаборативная фильтрация, метод косинусного сходства, метод корреляции Пирсона, спортзал, тренировка, спорт*

КОЛЛАБОРАТИВТІ ІРІКТЕУГЕ НЕГІЗДЕЛГЕН ЖАТТЫҒУЛАРДЫҢ ЖЕКЕ ҰСЫНЫМДАР ЖҮЙЕСІ

Аңдатпа: *Бұл мақалада біз коллаборативті іріктеу арқылы жаттығу ұсынымдарының жеке жүйесін іске асырудың бірнеше тәсілдерін іріктеп қарастырамыз. Негізгі бағыттар таңдалып алынды, жадымен жұмыс істеуге негізделген сүзу, онда деректермен тікелей жұмыс істейміз және өзімізге қажетті байланыстарды бөліп тастауға тырысамыз.*
Аталған Алгоритмдер тізіміне косинустық ұқсастықтар әдісі және Пирсон корреляция әдісі кіреді. Бірінші жағдайда біз әрбір пайдаланушыны n-өлшемді вектор ретінде аламыз, мұнда N-қарастырылатын жаттығулардың саны. Бұдан әрі екі нақты пайдаланушы арасындағы ұқсастықты олардың екі векторы арасындағы бұрыштың косинусы ретінде санаймыз. Бізге бұл тәсіл өте тиімді нәтиже берді, бірақ бос матрицаның жасушалары нәтижеге қатты әсер етті, өйткені бұл әдіс олармен нашар жұмыс істейді. Пирсон корреляциясы жағдайында юзерлер арасындағы оң немесе теріс трендтерді табуға және бос ұяшықтар үшін мәндерді болжау үшін пайдаланылатын корреляция коэффициентін есептеуге тырысамыз.
Мақаланың соңғы нәтижесі жоғарыда айтылған барлық әдістерді салыстырып, әр адамның артықшылықтары мен кемшіліктері туралы толық мағлұмат береді. Салыстыру RMSE және MAE сияқты метрикаларды есептеу арқылы жасалды.

Түйінді сөздер: *ұсыныстар жүйесі, коллаборативті іріктеу, косинустық ұқсастық әдісі, Пирсон корреляция әдісі, спорт зал, жаттығу, спорт*

Introduction

Collaborative filtering is a technique to filter out some items from given dataset, so they might be liked by user, based on user’s interactions with items.

First of all, we should take into account that this technique ignores all features of users and items it is considering. The reason for that is our algorithms main idea. It works by searching groups of similar users, not by their age or gender, but by their collaborations with items. For example, two users liked ten same items, then these be similar user despite their huge age difference etc.

The second step, after detecting groups, is predicting empty interactions in matrix, based on results of similar users. Our article contains many various approaches on finding similarities and predicting collaborations.

Main purpose of this article is to describe steps of implementing personalized training recommendation system based on collaborative filtering. Here we considered some different approaches from memory and model based filtering, such as cosine similarity, Pearson correlation and matrix factorization.

In addition, we have compared all algorithms considered in this article and provided results and their interpretations. By results we mean accuracy of techniques measured using such approaches as Root Mean Square Error (RMSE), Mean Absolute Error (MAE) etc.

All algorithms were implemented in Python programming language using such machine learning libraries as sci-learn kit, sci-py, surprise-py etc.

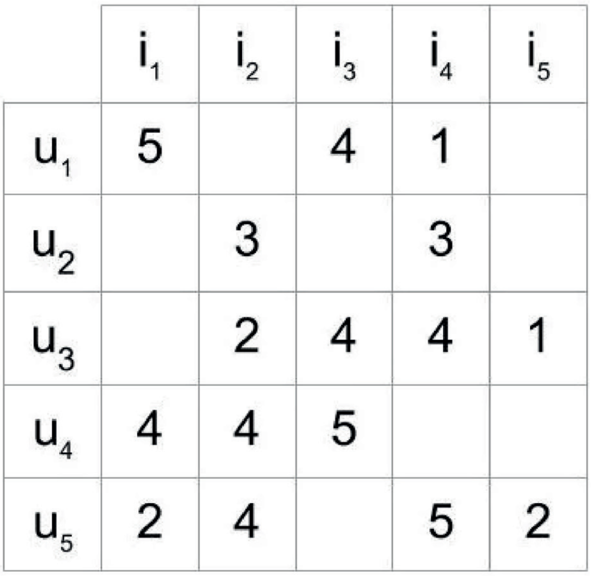
The Dataset

For our experiments we took data collected at Kazakhstani startup 1Fit which offers united subscription that lets user attend at any gym/

studio which is connected to the partner network. Data collected among users in Almaty and Nur-Sultan during period of 2018 – 2020 years.

Dataset itself contains collaborations between users and gyms in 2 types, first is rating from 1 to 5 indicating level of satisfaction and second is number of visits by user to some gym. We've divided dataset to 2 parts by their types and represented them as matrices, where one axis represent users and another represent gyms. Both of them are sparse and mostly consist of zeros, because there is no collaboration between user and given gym. Dataset contains more than 100k visits and 5k rating among 160 gyms and 5k users.

Moreover, user can rate a gym only after attending some of its trainings.



	i_1	i_2	i_3	i_4	i_5
u_1	5		4	1	
u_2		3		3	
u_3		2	4	4	1
u_4	4	4	5		
u_5	2	4		5	2

Fig.1. – Rating matrix sample

Memory based

This category of collaborative filtering includes algorithms which applies their statistical techniques to the entire dataset to calculate predictions.

Our aim in this methods is to predict **R** which is rating that **U** would rate some training **T.** This can be achieved by making two steps:

1. Find **K** most similar users to user **U**
2. Predict **R** based on ratings that **K** users have rated training **T**

Let's start from first step and consider case where we have only 2 trainings and 4 users. All of these user have rated both training and we have dataset like this.

1. User **A** = [1, 2]
2. User **B** = [3, 4]
3. User **C** = [5, 5]
4. User **D** = [2, 4]

Now let's visualize our dataset on graph:

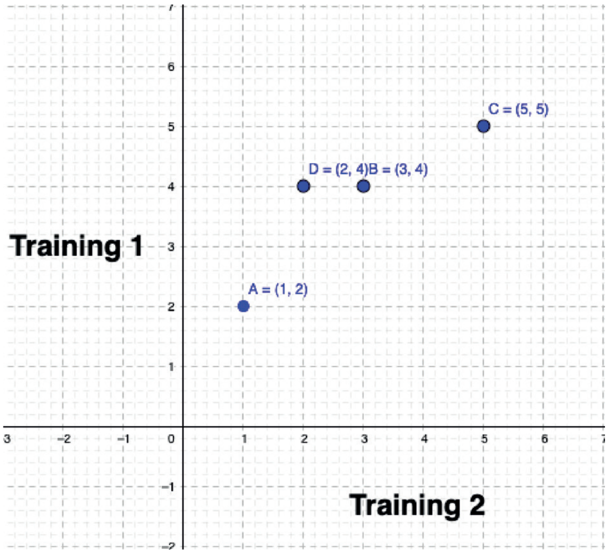


Fig.2. – Plot sample dataset as dots

Now, we can consider euclidean distance between each 2 users as their similarity. Calculating distance from **D** to each other gives as:

5. dist(**D**, **A**) = 2.5
6. dist(**D**, **B**) = 1
7. dist(**D**, **C**) = 2.23

Here we can see that closes user to **D** is **B** without any calculation, but what about the second closest user to **D**? By using euclidean distance we can conclude it as **C**. However, **C** has rating [5,5] whereas **A** is [1,2] and user **A** has rated each training as twice lower as user **D** did. Therefore, considering **A** might be better decision. How can we find relationships like this? Let's draw our graph converting dots to vectors:

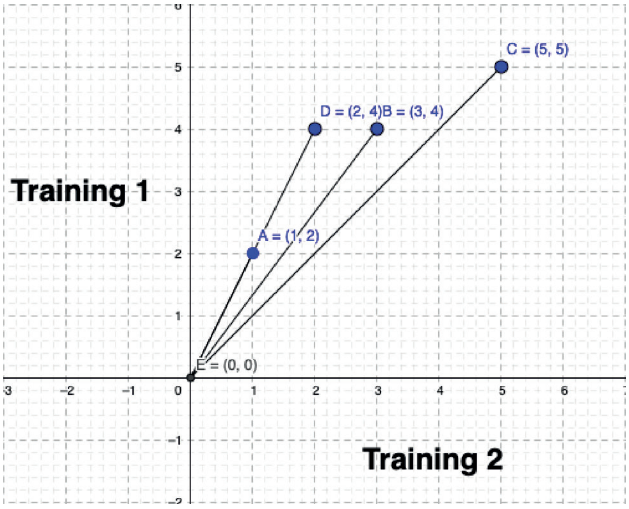


Fig.3. – Plot sample dataset as vectors

Now we can use angle difference as similarity coefficient instead of euclidean distance. The cosine of an angle is a function that decreases from 1 to -1 as the angle increases from 0 to 180. Therefore, it is convenient to use cosine of angle as similarity coefficient. This method is called cosine similarity.

Notice that in this case user **A** and **D** considered as 100% similar user, but they have different rating. Therefore, this isn't good enough for us and we have to move forward. The next step we can do is to subtract mean rating , which is mean rating for user **i,** from each users rating **i** and now we have:

1. User **A** has ratings [-1, 1]
2. User **D** has ratings [-0.5, 0.5]

Then, now mean value for user **A** and **D** is 0. After all, if now we adjust all user vectors this way, then mean off all dataset would be zero and using zero instead off empty matrix cell is now considered an mean. This method is called **centered cosine similarity or Pearson correlation.** Now we have treated different behaviors among user such as critic, tough raters, always 5 raters etc.

Second step was to calculate rating **R** based on **K** most similar users opinion. This can be done using formula:

$$R = \sum_{i=1}^{K} R_i / K$$

However, here don't consider coefficient of similarity and this might badly impact results, so it is better to use another formula where S_i is similarity of user **i** to considered user **U**:

$$R = (\sum_{i=1}^{K} R_i * S_i) / \sum_{i=1}^{K} S_i$$

In the above formula, every rating is multiplied by the similarity factor of the user who gave the rating. The final predicted rating by user **U** will be equal to the sum of the weighted ratings divided by the sum of the weights.

Implementation

Firstly, we apriori concluded that euclidean would give as bad results, because it is just learning example approach, but anyway we have run it on our dataset. For each case we run on 2 different matrices, where first is rating matrix and second is visits one. Before we started our test, we have removed unnecessary rows and column such as the user who attended less than 5 visits and who rated less than 3 of them. This optimized size of our matrices and made them less sparse. Then, after any test we have validated our results calculating such measures as RMSE and MAE. For validation we have used cross validation approach called Leave One Out.

Our results for euclidean distance were terrible bad. After 4 iterations we have got average *RMSE = 2.1019291126840174* and *MAE = 0.9680761932699417* for rating matrix. This is very bad, because rating values are between 1 and 5, then we can say that RMSE is really big for this case. For the test based on second dataset, we've similarly bad result as *RMSE = 9.532998814886966* and *MAE = 3.9483169498439312.*

In case of cosine similarity results was much better:

1. Rating: *RMSE = 0.906446677088667* and *MAE = 0.5713206807050449*
2. Visits: *RMSE = 3.236543673288123* and *MAE = 1.8913556812351201*

Pearson correlation gave as the best result that we are going to release on production someday:

1. Rating: *RMSE = 0.702423423012423* and *MAE = 0.3723641272356261*
2. Visits: *RMSE = 2.736543673223123* and *MAE = 1.3413551231232232*

Conclusion

We have considered different algorithms for collaborative filtering for our datasets based on real data from real life application. Each approach has its own cons and pros, so for collaborative filtering at all main disadvantage that this method doesn't take into account any features of user or items. It hardly depends on interactions, so new users or new items in set which has no collaboration yet would get bad predictions. The main advantage is that these methods doesn't require much performance and memory to implement.

REFERENCE

1. Francesco Ricci and Lior Rokach and Bracha Shapira (2011), *Introduction to Recommender Systems Handbook*, 1-35
2. Ferrari Dacrema, Maurizio; Cremonesi, Paolo; Jannach, Dietmar (2019), *Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches*, 101–109
3. He, Xiangnan; Liao, Lizi; Zhang, Hanwang; Nie, Liqiang; Hu, Xia; Chua, Tat-Seng (2017), *Neural Collaborative Filtering*,173–182
4. Fleder, Daniel; Hosanagar, Kartik (2009), *Blockbuster Culture's Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity*, 697–712.
5. Shi, Yue; Larson, Martha; Hanjalic, Alan (2014), *Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges*, 1–45