IRSTI 28.23.29

A. Kim[1]
[1]Suleyman Demirel University, Kaskelen, Kazakhstan

RECOMMENDER SYSTEM DEVELOPMENT

Abstract. Over the last few decades recommender systems have taken more and more place in industry. Recommender systems are algorithms aimed at suggesting relevant products to users. From suggesting to people the goods that could interest them to suggesting to users the web content matching their preferences, recommender systems are today unavoidable in our daily online journeys. Recommender systems are really important in all industries as they can bring real economic returns or can be competitive advantage.

In this work we provide an approach for increasing click-through rate by suggesting relevant products to users. The algorithm behind the recommender system described in this study is based on matrix factorization.

Keywords: recommender systems, matrix factorization, implicit feedback, explicit feedback.

***

Аннотация. В последние несколько десятилетий рекомендательные системы занимают все больше места в индустрии. Рекомендательные системы - это алгоритмы, предназначенные для предложения пользователям наиболее релевантных продуктов. Начиная с предложения людям товаров, которые могут их заинтересовать, и заканчивая предложением пользователям контента, соответствующего их предпочтениям, рекомендательные системы сегодня неизбежны в наших ежедневных онлайн-сессиях. Системы рекомендаций действительно важны во всех отраслях, поскольку они могут принести реальные экономические выгоды или могут стать конкурентным преимуществом.

В этой работе мы предлагаем подход, позволяющий повысить показатель кликов, предлагая пользователям наиболее релевантные товары. Алгоритм системы, описанной в этой работе, основан на матричной факторизации.

Ключевые слова: рекомендательная система, матричная факторизация, неявная обратная связь, явная обратная связь.

***

Андатпа. Соңғы бірнеше онжылдықта өнеркәсіпте ұсынушы жүйелер көбірек қолданысқа енгізілуде. Пайдаланушыларға оларды қызықтыратын элементтерді ұсынудан бастап, олардың қалауына сәйкес қажетті мазмұнды ұсынғанға дейін, ұсынушы жүйелер біздің күнделікті тіршілігіміздің бірден бір бөлігіне айналғаны сөзсіз. Ұсынушы жүйелер дегеніміз пайдаланушыларға тиісті элементтерді ұсынуға арналған алгоритмдер. Ұсынушы жүйелер барлық салада өте маңызды, өйткені олар үлкен табыс әкелуі мүмкін немесе бәсекелестерден айтарлықтай ерекшеленуге мүмкіндік береді.

Бұл жұмыста біз пайдаланушыларға тиісті тауарларды ұсына отырып, шерту трафигін арттыру тәсілін ұсынамыз. Алгоритм матрицалық жіктеуге негізделген.

Түйін сөздер: ұсынушы жүйелер, матрицалық жіктеу, анық емес кері байланыс, анық кері байланыс

Introduction

Today, there are many sites that provide any content, such as news, blogs, music and movies. Each of them contains a huge amount of information, but not all of it may be interesting to a particular visitor of the site. Recommender systems are used to select content that will be useful to a specific user. Unlike search engines, in order to get an answer, the recommendation system does not require an explicit request. The user is invited to evaluate some objects from the collection and, based on his estimates, assumptions are built and the results closest to them are returned. Recommender systems are very popular now since they significantly reduce the time it takes to find useful information.

The task of the recommender system is to inform the user about a product that he may be most interested in at a given time. The client receives information, and the service makes money on the provision of quality services. Services are not necessarily direct sales of the goods offered. The service can also earn on commissions or simply increase user loyalty, which then translates into advertising and other income.

Depending on the business model, recommendations can be its basis, as, for example, with TripAdvisor, or can be just a convenient additional service (such as, for example, in some online clothing store), designed to improve the Customer Experience and make catalog navigation more comfortable.

Personalization of online marketing is an obvious trend of the last decade. According to McKinsey, 35% of Amazon's revenue or 75% of Netflix's revenue comes from recommended products, and this percentage is likely to grow. Recommender systems are about what to offer the client to make him happy. To illustrate the whole variety of recommendation services, we will give a list of the main characteristics with which you can describe any recommendation system.

-Subject of recommendation - what is recommended.

There is a lot of variety here - it can be goods (Amazon, Ozon), articles (Arxiv.org), news (Surfingbird, Yandex.Zen), images (500px), videos (YouTube, Netflix), people (Linkedin, LonelyPlanet), music (Last.fm, Pandora), playlists, and more. In general, you can recommend anything.

-The purpose of the recommendation is why it is recommended.

For example: buying, informing, training, making contacts.

-The context of the recommendation is what the user is doing at this moment.

For example: watching goods, listening to music, talking with people.

-The source of the recommendation is who recommends:

-audience (average restaurant rating on TripAdvisor)

-interested users

-expert community (sometimes when it comes to a complex product, such as, for example, wine).

-The degree of personalization.

-Non-personal recommendations - when you are recommended the same as everyone else. They allow targeting by region or time, but do not take into account your personal preferences.

-A more advanced option is when recommendations use data from your current session. You have looked at several products, and at the bottom of the page you are offered similar ones.

-Personal recommendations use all available information about the client, including the history of his purchases.

-Transparency.

People trust the recommendation more if they understand exactly how it was received. So there is less risk of running into "unscrupulous" systems that promote paid goods or put more expensive goods higher in the ranking. In addition, a good recommender system itself should be able to deal with purchased reviews and sales cheats.Manipulations, by the way, are also unintentional. For example, when a new blockbuster is released, the first thing the fans go at it is, accordingly, the rating can be very high for the first couple of months.

-The format of the recommendation.

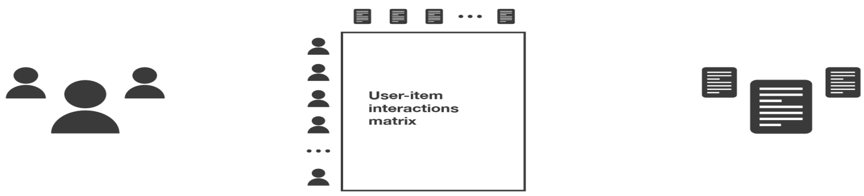
This can be a pop-up window, a sorted list appearing in a specific section of the site, a ribbon at the bottom of the screen, or something else.

-Algorithms

Despite the many existing algorithms, they all boil down to several basic approaches, which will be described later. The most classical are the algorithms.

Summary-based (non-personal), Content-based (models based on the product description), Collaborative Filtering (collaborative filtering), Matrix Factorization (methods based on matrix decomposition) and some others.

At the center of any recommendation system is the so-called preference matrix. This is a matrix, on one axis of which all the clients of the service (Users) are set aside, and on the other, objects of the recommendation (Items). At the intersection of some pairs (user, item), this matrix is filled with ratings (Ratings) - this is a well-known indicator of the user's interest in this product, expressed on a given scale (for example, from 1 to 5).



Users	User-item interactions matrix	Items
suscribers	rating given by a user to a movie (integer)	movies
readers	time spent by a reader on an article (float)	articles
buyers	product clicked or not when suggested (boolean)	products
	...	

Users usually evaluate only a small part of the products that are in the catalog, and the task of the recommendation system is to summarize this information and predict the customer's attitude to other products about which nothing is known. In other words, you need to fill in all the empty cells in the image above.

People's consumption patterns are different, and new products do not have to be recommended. You can show repeated positions, for example, to replenish stock. According to this principle, two groups of goods are distinguished.

- Repeatable. For example, shampoos or razors that are always needed.
- Unique. For example, books or films that are rarely re-acquired.

If the product cannot be explicitly assigned to one of the classes, it makes sense to determine the permissibility of repeat purchases individually (someone goes to the store just for the sake of peanut butter of a certain brand, and for someone it is important to try everything that is in the catalog).

The concept of "interest" is also subjective. Some users only need things from their favorite category (conservative recommendations), while others, on the contrary, are more responsive to non-standard products or groups of products (risky recommendations). For example, video hosting can recommend the user only new episodes of his favorite series, and can periodically throw him new shows or even new genres. Ideally, you should choose a strategy for displaying recommendations for each client separately, using the modeling of a client category.

User ratings can be obtained in two ways:

-explicit ratings - the user puts the product rating, leaves a review, likes the page,

-implicit ratings - the user does not explicitly express his attitude, but an indirect conclusion can be drawn from his actions: he bought the product - that means he likes it, read the description for a long time - it means there is interest, etc.

Of course, explicit preferences are better - the user himself says that he liked it. However, in practice, not all sites provide the opportunity to express their interest explicitly, and not all users have the desire to do so. Both types of ratings are most often used at once and complement each other well.

Development of recommender system

Dataset:

-100000 unique user_id

-1500000 interactions

->45000 items , reduced to 600 (explained below)

-97.5 % sparcity

Feedback:

-implicit (buys)

-explicit (not interesting recommendations)

Databases:

-MySQL(transactions)

-PostgreSQL(recommendation and feedback)

Tools and technologies :

-Python(for modeling and processing)

-CRON(for scheduled tasks)

-SQL(for data manipulations)

-Google analytics(for monitoring)

Dataset is presented as interaction of users and deals. Deals are offers which can be bought by user. Every deal has several categories and subcategories , price and other characteristics. Every deal has limited lifetime ,so we cannot recommend deals which are already closed. In order to overcome this problem, we grouped similar deals into clusters using the following process.

Every deals is presented as one-hot encoded vector of subcategories and price categories .

At the output we have this table (table 1):

	category_1	category_2	category_3	category_4	category_5	category_6	category_7
item_1	1	1	0	0	0	0	1
item_2	0	0	1	1	0	0	0
item_3	1	1	0	0	0	0	1

Table 1. One-hot encoded deals

As we can see item_1 and item_3 have similar vector , so they will be grouped into one cluster. Number of clusters equals to number of unique possible combinations.

Now we recommend cluster instead of deals and fetch only active deals from clusters.

This operation reduced size of user-item interactions from (100000*45000 to 100000*600). After showing recommendations user can explicitly give us feedback if items are not relevant.

Matrix factorization

From mathematics we know that any matrix can be decomposed into the product of three matrices. But the assessment matrix is very sparse, 99% is commonplace. Singular Value Decomposition does not know what gaps are. We don't really want to fill them with an average value. And in general, we are not very interested in the matrix of singular values - we just want to get a hidden view of users and objects, which, when multiplied, will approximate the true rating. We can immediately decompose into two matrices.

What to do with missing values? We can ignore them. It turned out that we can train to approximate ratings by RMSE metric using SGD or ALS, ignoring omissions altogether. The first such algorithm is Funk SVD, which was invented in 2006 during the solution of the competition from Netflix.

Netflix Prize - a famous event that gave a strong impetus to the development of recommendation systems. The goal of the competition is to overtake the existing Cinematch recommendation system for RMSE by 10%. For this, a large dataset containing 100 million ratings was provided at that time. The task may not seem so difficult, but to achieve the required quality it was required to rediscover the competition two times - a solution was received only for 3 years of the competition. If it were necessary to obtain an improvement of 15%, perhaps this would not have been possible to achieve on the data provided.

The main idea of matrix factorization (Fig 1):

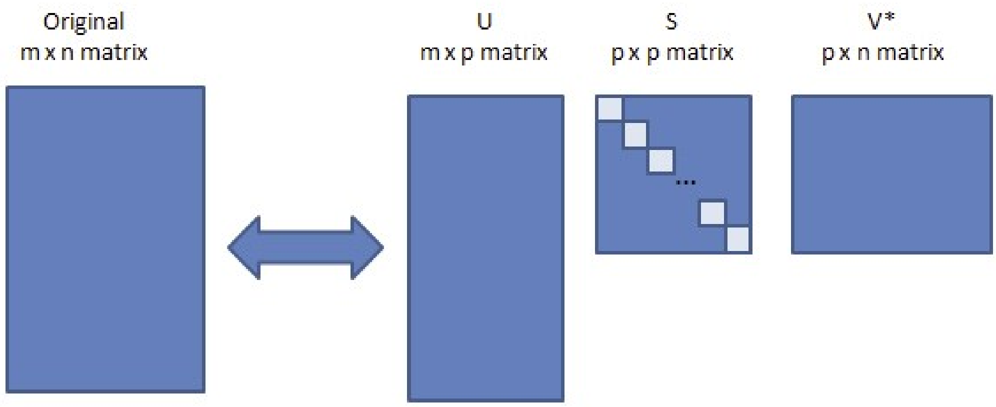


Fig 1 Matrix factorization

Loss functions

To use implicit feedback, we came up with appropriate teaching methods.

Bayesian personalized ranking

It is known what items the user interacted with. We assume that these are positive examples that he liked. There are still many items that the user has not interacted with. We do not know which of them will be of interest to the user and which are not, but we probably know that not all of these examples will turn out to be positive.

We make a rough generalization and consider the absence of interaction as a negative example.

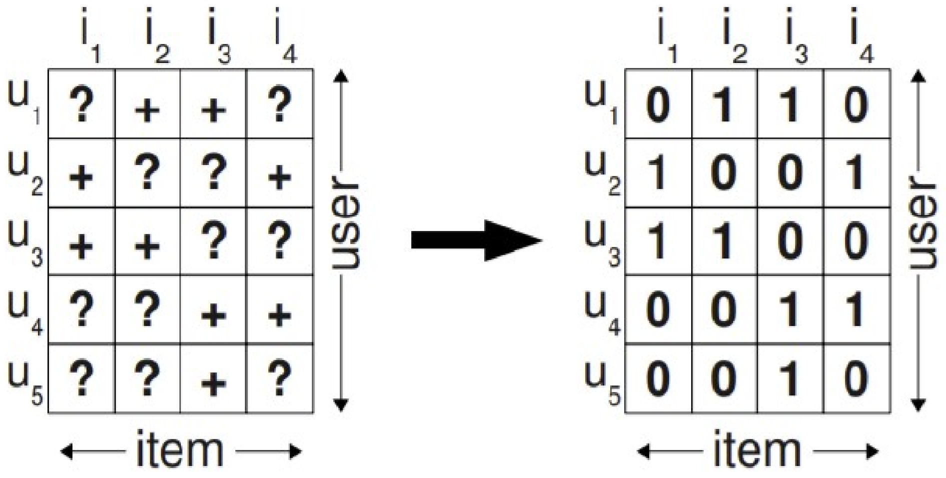


Fig. 2 Implicit feedback interaction matrix

We will sample triples {user, positive item, negative item} and punish the model if the negative example is rated higher than the positive.

$LBPR(u,i,j)=1-\sigma(\hat{r}_{ui}-\hat{r}_{uj})$

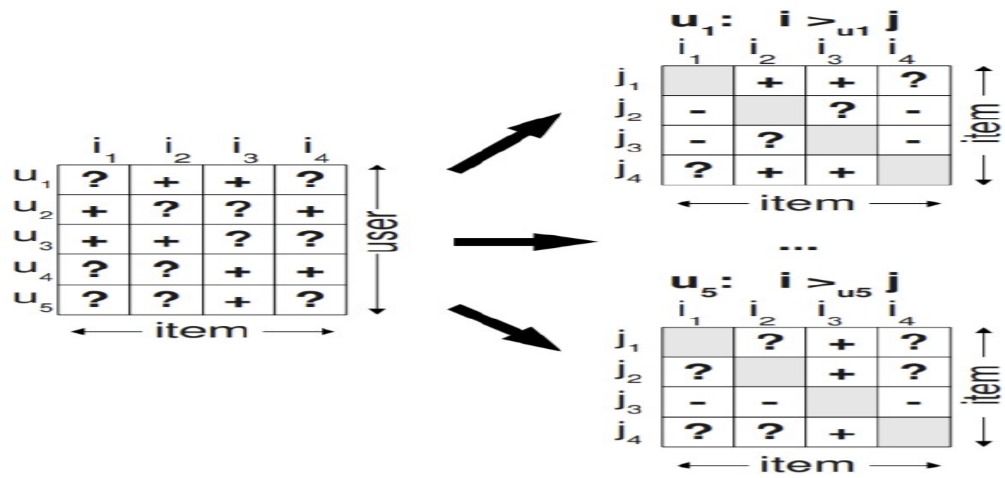


Fig. 3 Sample triples

Quality metrics

-Online evaluation

The most preferred way to evaluate the quality of the system is to directly check on users in the context of business metrics. This can be CTR, time spent in the system, or the number of purchases. But experiments on users are expensive, and we don't want to roll out a bad algorithm even to a small group of users, so they use offline quality metrics before online testing. In this work we used CTR.

-Offline evaluation

As quality metrics, ranking metrics are usually used. MAP@k is used in our case.

Let ***R*** be the set of recommended objects, ***P*** the set of objects that the user actually

likes.

Then :

Precision = (R and P) / R

Precision @ k - is the proportion of recommended items in the top-k set that are relevant

Precision@k = (# of recommended items @k that are relevant) / (# of recommended items@k)

Recall = (R and P) / P

Recall @ k is the proportion of relevant items found in the top-k recommendations

Recall@k = (# of recommended items @k that are relevant) / (total # of relevant items)

Rank	Correct?	Precision	Recall
1	True	1.0	0.2
2	True	1.0	0.4
3	False	0.67	0.4
4	False	0.5	0.4
5	False	0.4	0.4
6	True	0.5	0.6
7	True	0.57	0.8
8	False	0.5	0.8
9	False	0.44	0.8
10	True	0.5	1.0

Fig. 4 Precision and recall calculation

Results and conclusions

During this work recommender system was developed, tested and integrated.

Data was collected, organized and stored. Implicit and explicit feedback were used to improve final result.

The main metrics obtained during testing are : MAP@3 - 0.08, CTR - + 40%.

Cold start problem was solved by sampling users only with purchase history. Interaction matrix dimensionality was decreased by using clusters, instead of items directly.

This work can be improved by using behavioral features and adaptive learning for loss function.

References

1 Maciej Kula. Metadata Embeddings for User and Item Cold-start Recommendations . - 2015. - CEUR Workshop Proceedings № 1448. P. 14-21.

2 Bokde, D., Girase, Sh., Mukhopadhyay, D. Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey. Procedia Computer Science. 49. DOI: 10.1016/j.procs.2015.04.237.

3 Koren, Y., Bell, R., Volinsky, Ch. Matrix factorization techniques for recommender systems. *Computer,* 42 (2009): pp. 30-37.

4 Mehta, R., Rana, K. A review on matrix factorization techniques in recommender systems. (2017). 269-274. DOI: 10.1109/CSCITA.2017.8066567.

5 Ilhami, M., Suharjito, S. Film Recommendation Systems using Matrix Factorization and Collaborative Filtering. *2014 International Conference on Information Technology Systems and Innovation, ICITSI 2014 - Proceedings.* DOI: 10.1109/ICITSI.2014.7048228.