УДК 631.15:33

THE USE OF MACHINE LEARNING IN RECOMMENDED SYSTEM IN THE E-COMMERCE

Y. Akhmer
phd doctoral student of the department of computer engineering of the International University of Information Technology

G.U. Bektemyssova
associate professor, department of computer engineering, International University of Information Technology

The Department of Computer Engineering
International Information Technology University, Almaty, Kazakhstan
E-mail: g.bektemisova@iitu.edu.kz

Y. Akhmer
Is PhD doctoral student of the Department of Computer Engineering of the International University of Information Technology His research interests include ML, data science, data analysis, natural language processing, predictive models etc.

G. Bektemyssova
Is a candidate of technical sciences; currently holding the positions of associate professor at International Information Technology University (IITU) and director of the IITU Department of Postgraduate Education. Her research interests include machine learning (ML), decision support systems, data science, big data etc..

Abstract. A filtering method is indispensable in a data-flooded environment. Recommended systems have made a massive step towards this aim, speeding up internet-based customer experience. Most of today's examples of artificial marketing intelligence are known as supervised learning, which varies from offering personalized specific products identifying the most valuable marketing strategies, to forecasting customer churn rate or customer life value, and building up a positive client base. Generally, different types of stored information are used to customize various dimensions or search results, demonstrate the most targeted advertising on the homepage, etc. Recommended systems make a profit by using suggestions to generate sales. Every other system can use different data from multiple sources to assess the usage patterns and discover similar trends which forecast future customers' purchases or preferences. It predicts interesting patterns and provides guidance based on the customer interest model. There seems to be, on the one side, a traditional recommendation system that proposes items based on various criteria of consumers or products such as product price, user information and etc., but on the other hand, we have also recommended systems incorporating deep learning methods, even if they have not yet been well investigated. This paper discusses various processes associated with implementing recommenders systems and numerous recommender approaches along with the analysis of those methods that can be used by different scholars across several papers. Implementation of collaborative filtering method and content-based filtering techniques is pretty much pointless, since most e-commerce shops are already using hybrid engines, which have proved to be more effective. In our research we have also incorporated the benefits and drawbacks of every approach. Finally, this paper also presents numerous difficulties and problems confronting recommenders in their application systems algorithms. In this paper, we initially present multiple best known types of recommended systems and concentrate on one part of the e- commerce recommendation and afterwards make their quantitative comparison. Recommender systems have taken a huge step towards this goal, greatly improving the user experience in the online environment.

Keywords: Recommendation Systems, E-commerce, Content-based Filtering, Collaborative Filtering, User-based Filtering.

Introduction.

The field of e-commerce is dominated by, personalization services, aimed at optimizing the site content for a specific consumer. The standard big data processing system includes an analysis of four parameters: data on a specific user, data on the entire population of users, information on the properties of the product and external factors. Based on the above criteria, the system automatically selects the most relevant products for the consumer, thereby improving the quality of service and sales.

Key players in the e-commerce industry, such as Amazon or AliExpress, prefer to use their own developments in the field of customizing the product range (for example, AWS). The statistical data show the effectiveness of these tools' implementation and improvement for working with Big Data in Internet commerce. Specifically, RichRelevance provides data on more than 10% sales growth and 300% investment efficiency for companies using BD solutions in their work. The Russian analogue of foreign systems - RetailRocket declares the possibility of increasing the online store sales by 10-50%. Thus, the application of Big Data technologies in e- commerce today is relevant and continues to evolve.

As of the beginning of 2019, the online trading market of the Republic of Kazakhstan was estimated at 287 billion tenge, manifesting the total 23.2% annual growth. The share of online trading in total trade amounted to only 2.9%, which, according to Nikolai Babeshkin, indicates a significant growth potential. The forecast for global online market growth is 11% per year. At the same time, the potential of Kazakhstan is quite high, given its level of Internet penetration. According to World Cellular Information Service -, in 2017 Kazakhstan had 76.4% of the Internet users. In this rating, Kazakhstan ranks second to the UK (94.8% of users), ahead of even the United States (76.2%), Poland (76%) and Russia (76%). An increase in the number of connections via smartphones in the republic add to the positive picture: by the end of 2018, there were 18.2 million, and 25.6 million are forecast by 2022. By this period, smartphones should account for 82% of the total number of mobile connections. According to the Digital Kazakhstan Association (DKA) experts, in 2022 the e-commerce market in Kazakhstan may be worth 928 billion tenge. That is, according to cautious estimates based on the global average growth, there will be a 6% increase.

Recommender systems are a large class of models whose goal is to increase business performance by providing relevant recommendations to the user in the right place, at the right time, and through the right communication channel.

Every day, millions of people are searching the Internet: someone is looking for movies or clothes, someone is looking for a car or a vacation package, and all users are united by one goal: to find what they need. If in the last century people learnt about the emergence of new goods from mailing lists, by now this process has been accelerated by dozens (or even hundreds) times due to the appearance of television and then the Internet.

And in recent decades, the use of machine learning algorithms has become one of the leading trends in improving a wide variety of seaвaccccccccc45кrch engines. As an addition to the process of independent search (among millions of names of various goods and services), recommendation systems began to predict what exactly would be interesting for this or that user. In the course of their work such recommendation algorithms have been constantly trained, adapted and transformed, to better understand the user, and as a result of their functioning, 50% or more of the recommended goods or services to some extent or another satisfy the users' search queries. This article provides an analysis of the principles of operation of the main methods for implementing recommender systems and metrics to evaluate their performance.

Related work

This section presents some related work that uses recommended system techniques in e-commerce and on online store websites.

The recommender system is described as a user- friendly decision-making strategy in advanced data environments [1]. The recommendation system was also classified first from the point of view of e-commerce mostly as a method that allows people to interact through information data relating to the users' requirements and needs [2]. The recommendation system was considered as a technique of supporting and enhancing the social method of creating options via the use of recommendations from others since there is no adequate specific understanding or knowledge of alternatives [3]. The recommender systems (Jannach et al. 2010) leave reviews (options, strategies) relevant to the user. The recommender systems address the issue of information overload that customers are likely to experience by supplying the latter with individualized, unique subject matter and delivery recommendations.

Recommended techniques primarily boil down to two main methods: collaborative filtering and content-based filtering. Collaborative Filtering (Konstan et al. 1997) uses the view of customers with similar choices, while content-based filtering (Pazzani and Billsus 1997) is centered on a comparison of the information of the already purchased products with the new products that could possibly be recommended to the customer. Certain basic recommendations are knowledge-based recommendations, group recommendation systems, and hybrid recommendations. Knowledge-based advising systems (Felfernig et al. 2015) focus on knowledge acquisition, guidelines or limitations on a product array, user behavior and recommendations metrics (i.e. which product should be suggested in this or that scenario). Group recommendation systems (Felfernig et al. 2018; Masthoff 2011) measure recommendations in which the whole cohort ought to be satisfied with the recommendation. A hybrid recommendation (Burke 2002) integrates fundamental recommendations to help make up for the flaws of the samples treated.

Several methods of developing recommendation systems have recently been introduced which use collaborative filtering, content-based filtering or hybrid filtering [11], [12], [13]. The collaborative filtering method seems to be the most complete and perhaps the most widely used. Collaborative filtering recommends items by classifying certain customers with similar preferences; it uses everyones viewpoint to recommend products to the active user. Collaborative recommended systems have been developed in various application domains. The system then proposes some similar products or services on-line as per the customer's previous purchases. From the other side, content-based methods align information assets with user preferences. Content-based filtering methods typically center their assumptions on customer data and dismiss commitments from all other customers, like in the particular instance of collaborative techniques [14], [15].

Despite the success of these two filtering techniques, several limitations have been identified. Some of the problems with content-based filtering techniques are associated with limited content analysis, overspecialization and sparsity of data [16]. Also, collaborative approaches generate cold-start, sparsity and scalability problems Based on the effectiveness of such two filtering methods, a number of constraints have been recognized. A few of the difficulties related to content-based filtering methods include restricted comparative study, over-specialization and data sparseness [16]. Collaborative techniques also have issues with cold-start, sparsity and scalability. Such issues usually lower the productivity of the recommendations. Hybrid filtering, which integrates multiple filtering methods in a variety of ways to improves the efficiency and productivity of the recommended systems, has indeed been suggested [17], [18], in order to alleviate some more of the current challenges. All such methods merge multiple or more filtering methods in order to manipulate their strong points while at the same time balancing their respective weaknesses [19]. Based mostly on their activities, they could be categorized into a weighted hybrid, mixed hybrid, switching hybrid, feature-combination hybrid, cascade hybrid, feature-augmented hybrid and meta-level hybrid [20]. Cunningham et al. [21] developed a clear and easy technique for integrating content-based and collaborative filtering.

Recommended system phases: theoretical review

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 the 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. Transparency is one of the important characteristics of the system. People trust the recommendation more if they understand exactly how it has bee 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, it is the first thing the fans go at, accordingly, the rating can be greatly overestimated for the first couple of months. This section is devoted to the description of algorithms which are also an integral part of any recommendation system. Despite the many existing algorithms, they all boil down to several basic approaches, which will be analyzed below. The most classical algorithms include summary-based (non-personal), content-based (models based on product description), collaborative filtering, matrix factorization (methods based on matrix decomposition), hybrid and some others.

A. Non-personalized recommendations

Let us consider non-personalized recommendations because they are the easiest to implement. Here the potential interest of the user is simply determined by the average rating of the product: “Everyone likes it, so you will like it.” Most of the services work on this principle when the user is not logged in to the system, for example, the same TripAdvisor.

1) Cold start problem

A cold start is a typical situation when enough data have not yet been accumulated for the recommender system to work correctly (for example, when a product is new or just rarely bought). If the average rating is calculated by the estimates of only three users (Alice, Bob and Eve), such an assessment will clearly not be reliable, and users understand this. In such situations, ratings are often artificially adjusted.

The first way is to show not the average value, but the smoothed average (Damped Mean). The meaning is this: with a small number of ratings, the displayed rating is more inclined to a certain safe “average” indicator, and as soon as a sufficient number of new ratings is gathered, the “average” adjustment ceases to work.

Another approach is to calculate confidence intervals for each rating. Mathematically, the more estimates, the less variation of the average and, therefore, more confidence in its correctness. And as a rating you can display, for example, the lower boundary of the interval (Low CI Bound). At the same time, it is clear that such a system will be quite conservative, with a tendency to underestimate ratings for new products (unless, of course, this is a hit).

Since estimates are limited to a certain scale (for example, from 0 to 1), the usual method of calculating the confidence interval is poorly applicable here: because of the distribution tails that go to infinity and the symmetry of the interval itself. There is an alternative and more accurate way to calculate it - Wilson Confidence Interval. In this case, asymmetric intervals are obtained, as shown in Figure 1.

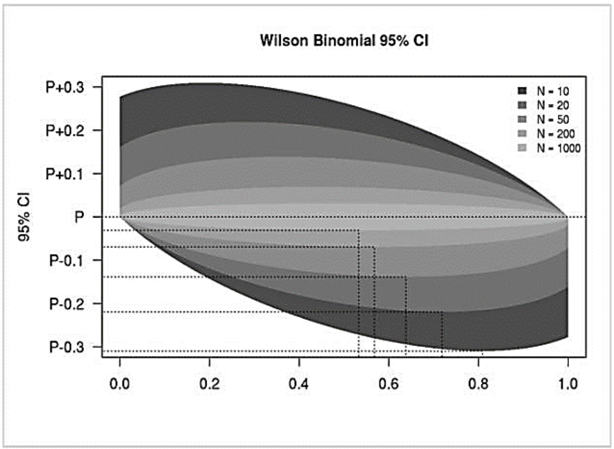

Figure 1. Wilson Confidence Interval

In the figure above, the horizontal rating of the average rating is plotted, and the vertical is the spread around the average. Different sizes of the sample are highlighted in color (obviously, the larger the sample, the smaller the confidence interval).

The cold start problem is just as relevant for non- personalized recommendations. The general approach here is to replace what cannot be counted at the moment with various heuristics (for example, replace it with an average rating, use a simpler algorithm, or not use a product at all until data is collected).

2) *Relevance of recommendations*

In some cases, it is also important to consider the “freshness” of the recommendation. This is especially true for articles or forum posts. Fresh entries should hit the top more often. For this, correction factors (damping factors) are used. Below are a couple of formulas for calculating the ranking of articles on media sites (Figure 2).

Example of rating calculation in Hacker news magazine:

$$Rank = \frac{(U - D - 1)^{0.8} \times P}{T^{1.8}}$$

Figure 2. Example of rank calculation

where U = upvotes, D = downvotes, and P (Penalty) is an additional adjustment for the implementation of other business rules.

Not all elements are equally significant: for example, allied words, obviously, do not carry any payload. Therefore, when determining the number of matching elements in two vectors, all measurements must first be weighed by their significance. This task is solved by the TF-IDF transformation well known in Text Mining as shown in Figure 4, which assigns more weight to rarer interests. The coincidence of such interests is more important in determining the proximity of two vectors than the coincidence of popular ones.

Rating calculation in Reddit:

$$Rank = \log_{10}\left(\max\left(1, U - D\right)\right) - \frac{|U - D|T}{const}$$

Figure 3. Example of Reddit calculation

where U = the number of votes in favor, D = the number of votes against, T = the time of recording. The first term estimates the “recording quality”, and the second makes a correction for time.

Obviously, a universal formula does not exist, and each service invents the formula that best solves its problem - it is verified empirically.

B. Content-based recommendations

Personal recommendations suggest the maximum use of information about the user himself, primarily about his previous purchases. One of the first approaches used for the purpose was the content-based filtering approach. In the framework of this approach, the description of the product (content) is compared with the interests of the user obtained from his previous ratings. The more the product meets these interests, the higher is the evaluated potential interest of the user. The obvious requirement here is that all products in the catalog should have a description.

Historically, the subject of content-based recommendations has often been goods with an unstructured description: films, books, articles. Such signs may be, for example, text descriptions, reviews, casts and more. However, nothing prevents the use of ordinary numerical or categorical signs.

Unstructured features are described in a way typical of text - vectors in the word space (Vector-Space model). Each element of such a vector is a feature that potentially characterizes the user's interest. Similarly, a product is a vector in the same space.

As the user interacts with the system (say, he buys films), the vector descriptions of the goods purchased by him are combined (summed and normalized) into a single vector and, thus, a vector of his interests is formed. Further, it is enough to find a product whose description is closest to the vector of interests, i.e. solve the problem of finding n nearest neighbors.

$$W_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

Figure 4. TF-IDF transformation

The TF-IDF principle here is equally applicable to ordinary nominal attributes, such as, for example, genre, director, language. TF – is a measure of the importance of the attribute for the user, IDF - a measure of the "rarity" of the attribute.

There is a whole family of similar transformations (for example, BM25 and similar ones), but in substance they all repeat the same logic as TF-IDF: rare attributes should have more weight when comparing products. Figure 5 below illustrates how the weight of TF-IDFs depends on TF and IDF. The nearest horizontal axis is DF: attribute frequency among all products, the far horizontal axis is TF: user’s attribute frequency logarithm.

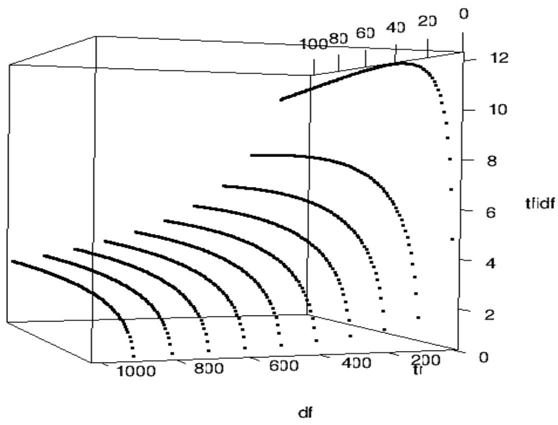

Figure 5. The weight of TF-IDFs

Some points to consider when implementing.

- When forming a vector-space presentation of a product, instead of individual words, you can use shingles or n-grams (consecutive pairs of words, triples, etc.). This will make the model more detailed, but more data will be needed for training.
- In different places of the product description, the weight of the keywords may differ (for example, the description of the film may consist of a title, a brief description and a detailed description).
- Product descriptions from different users can be weighted differently. For example, we can give more weight to active users who have many ratings.
- Similarly, you can weigh a product. The higher is the average rating of an object, the greater is its weight (similar to PageRank).
- If the product description allows links to external sources, then you can get confused and analyze all third-party information related to the product.

It can be seen that content-based filtering almost completely repeats the query- a document matching mechanism used in search engines such as Yandex and Google. The only difference is in the form of a search query - here is a vector describing the interests of the user, and the keywords of the requested document. When search engines begin to add personalization, the distinction is erased even more. As a measure of the proximity of two vectors, the cosine distance is most often used (Figure. 6).

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

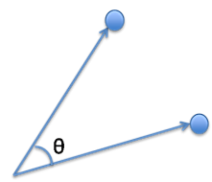

Figure 6. A measure of proximity

When a new assessment is added, the vector of interests is updated incrementally (only for those elements that have changed). When recounting, it makes sense to give new estimates a little more weight, since preferences may vary.

C. Collaborative filtering (User-based option)

This class of systems began to develop actively in the 90s. As part of the approach, recommendations are generated based on the interests of other similar users. Such recommendations are the result of the “collaboration” of many users. The classic implementation of the algorithm is based on the principle of k nearest neighbors. On the fingers - for each user, we look for k most similar to him (in terms of preferences) and supplement the information about the user with data known about his neighbors. So, for example, if it is known that your interest neighbors are delighted with the film "Blood and Concrete", and you haven’t watched it for some reason, this is a great reason to offer you this film for Saturday viewing (Figure. 7).

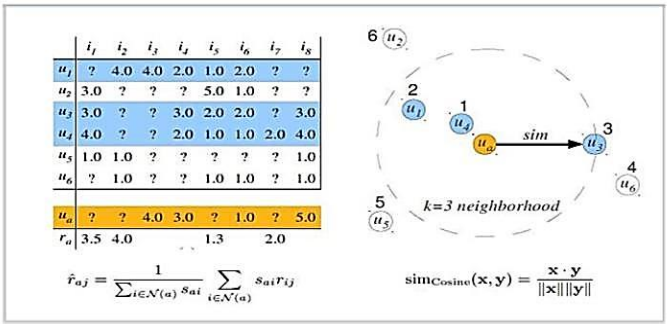

Figure 7. An example of Collaborative method

The figure above illustrates the principle of the method. In the preference matrix, the user for which we want to determine the ratings for new products (question marks) is highlighted in yellow. Three of his closest neighbors are highlighted in blue.

“Similarity” of interests is in this case a synonym for their “correlation” and can be considered in many ways (in addition to Pearson correlation, there is also a cosine distance, a Jacquard distance, a Hamming distance, etc.).

The classical implementation of the algorithm has one obvious minus - it is poorly applicable in practice due to quadratic complexity. Indeed, like any method of the nearest neighbor, it requires the calculation of all pairwise distances between users (and there may be millions of users). It is easy to calculate that the complexity of calculating the distance matrix will be O (n^2, m) where n is the number of users and m is the number of products. With a million users, a minimum of 4TB is required to store the distance matrix raw.

This problem can be partially solved by purchasing a high- performance iron. But if you approach wisely, it is better to introduce corrections into the algorithm:

- update distances not with every purchase, but with batches (for example, once a day),
- do not recalculate the distance matrix completely, but update it incrementally,
- opt for iterative and approximate algorithms (for example, ALS).

For the algorithm to be effective, it is important that a few assumptions are fulfilled.

- People’s tastes do not change with time (or change, but in the same manner for everyone).
- If people's tastes coincide, then they coincide in everything.

For example, if two clients prefer the same films, then they also like the same books. This often happens when the recommended products are homogeneous (for example, only films). If this is not so, then a couple of customers may well have the same food preferences, and political views be directly opposite - here the algorithm will be less effective.

The user's neighborhood in the preference space (his neighbors), which we will analyze to generate new recommendations, can be chosen in different ways. We can work with all users of the system in general, we can set a certain proximity threshold, we can select several neighbors randomly or take the n most similar neighbors (this is the most popular approach).

The authors of MovieLens as the optimal number of neighbors give figures of 30-50 neighbors for films and 25-100 for arbitrary recommendations. It is clear here that if we take too many neighbors, we will get more chance of random noise. And vice versa, if we take too little, we will get more accurate recommendations, but fewer products can be recommended. An important stage in the preparation of data is the normalization of estimates.

1) Data standardization (scaling)

Since all users evaluate differently - someone puts five in a row, and you rarely expect four from someone - it’s better to normalize the data before calculating, i.e. lead to a single scale so that the algorithm can correctly compare them with each other.

Naturally, the predicted estimate will then need to be translated into the original scale by the inverse transformation (and, if necessary, round to the nearest integer).

There are several ways to normalize:

- centering (mean-centering) - we simply subtract their average rating from the user's ratings,
- standardization (z-score) - in addition to centering, we divide its assessment by the standard deviation of the user, relevant only for non-binary matrices (after the reverse conversion, the rating may go beyond the scale (i.e., for example, 6 on a five-point scale), but such situations are quite rare and are solved simply by rounding towards the nearest acceptable rating),
- double standardization – firstly, we normalize user ratings, secondly- product ratings.

If the movie “The Best Movie” has an average rating of 2.5, and the user gives it 5, then this is a strong factor indicating that such films are clearly to his taste. The "similarity" or correlation of the preferences of two users can be considered in different ways. In fact, we just need to compare two vectors and list the most popular correlations.

1. Pearson correlation is a classical coefficient, which is quite applicable when comparing vectors (Figure. 8).

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

Figure 8. Pearson correlation coefficient

Its main disadvantage is that when the intersection is estimated to be low, the correlation can be high simply by accident.

To combat a randomly overstated correlation, you can multiply it by a factor of 50 / min (50, Rating intersection) or any other damping factor, the influence of which decreases with the increasing number of ratings.

2. Spearman correlation

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2 - 1)}.$$

Figure 9. Spearman correlation coefficient

The main difference is the rank coefficient, i.e. it works not with absolute ratings, but with their serial numbers. In general, it gives a result very close to Pearson's correlation.

3. Cosine distance

Another classic factor. If you look closely, the cosine of the angle between standardized vectors - is Pearson's correlation, calculated using the same formula:

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Figure 10. The similaity

It is called the cosine distance - because if two vectors are aligned (that is, the angle between them is zero), then the cosine of the angle between them is equal to one. Conversely, the cosine of the angle between perpendicular vectors is zero.

An interesting development of the collaborative approach are the so-called Trust-based recommendations, which take into account not only the proximity of people according to their interests, but also their “social” proximity and the degree of trust between them. If, for example, we see that on Facebook the girl periodically visits the page with the audio recordings of her friend, then she trusts her musical taste. Therefore, recommendations to the girl can completely mix new songs from the friend’s playlist.

2) Justification of recommendations

It is important that the user trusts the recommendation system, and for this it should be simple and understandable. If necessary, a clear explanation of the recommendation should always be available (in English terms).

As part of the explanation, it’s nice to show the product’s assessment of the neighbors, according to which attribute (for example, the actor or director), there was a coincidence, as well as display the confidence of the system in the assessment (confidence). In order not to overload the interface, you can put all this information into the “Tell me more” button. For instance: “You might like the movie ... and plays there.”, "Users with similar musical tastes rated the album 4.5 out of 5."

D. Collaborative filtering (Item-based option)

The Item-based approach is a natural alternative to the classical User-based approach described in the first part, and repeats it almost completely, except for one point - it applies to the transposed preference matrix, i.e. looks for related products, not users.

Let me remind you that the user-based collaboration filtering (user-based CF) searches for each client for a group of customers most similar to him (in terms of previous purchases) and averages their preferences. These average preferences serve as recommendations for the user. In the case of commodity collaborative filtering (item-based CF), the closest neighbors are searched for on the set of goods - columns of the preference matrix, and averaging occurs precisely on them.

Indeed, if the products are meaningfully similar, then most likely they are either liked or not liked at the same time. Therefore, when we see that the valuations of two products are strongly correlated, this may indicate that they are analogous goods.

Advantages of the item-based approach over the user-based one:

- When there are a lot of users (almost always), the task of finding the nearest neighbor becomes poorly computable. For example, for 1 million users, you need to calculate and store ~ 500 billion distances. If you encode the distance with 8 bytes, this results in 4TB for the distance matrix alone. If we do an Item- based search, then the complexity of the calculations decreases from O(N2n) to O(n2N), and the distance matrix has a dimension no longer than 1 million per 1 million but, for example, 100 per 100 by the number of products.
- The proximity rating of products is much more accurate than the proximity rating of users. This is a direct consequence of the fact that there are usually many more users than goods, and

therefore there is much less the standard error in calculating the correlation of good. We just have more information to draw a conclusion.

- In the user-based version, user descriptions are usually very sparse (there are a lot of products, few ratings). On the one hand, this helps to optimize the calculation - we multiply only those elements where there is an intersection. But on the other hand – no matter how many neighbors you take, the list of goods that you can eventually recommend is very small.
- User preferences may change over time, but the item description is much more stable.

The rest of the algorithm almost completely repeats the user- based option: the same cosine distance as the main measure of proximity, the same need for data normalization. The number of neighboring goods N is usually chosen in the region of 20.

Due to the fact that the correlation of products is considered on a larger number of observations, it is not so critical to recalculate it after each new assessment, and you can do this periodically in the battle mode.

Several possible improvements to the algorithm:

- An interesting modification is to consider the “similarity” of products not as typical cosine distances, but by comparing their content (content- based similarity). If at the same time the user preferences are not taken into account in any way, such filtering ceases to be “collaborative”. Moreover, the second part of the algorithm – obtaining averaged estimates – does not change in any way.
- Another possible modification is to weigh users when calculating item similarity. For example, the more users make ratings, the more weight they have when comparing two products.
- Instead of simply averaging estimates for neighboring products, weights can be selected by doing a linear regression.

When using the item-based approach, recommendations tend to be more conservative. Indeed, the scatter of recommendations is less and therefore less likely to show non- standard products.

If in the preference matrix we use the product description view as a rating, then the recommended products are most likely to be analogues - products that are often viewed together. If we calculate the ratings in the preference matrix based on purchases, then most likely the recommended products will be accessories - goods that are often bought together.

E. Factorization Algorithms

It would be great to describe the interests of the user in “larger strokes." Not in the format “he loves films X, Y and Z”, but in the format “he loves modern Russian comedies”.

Besides the fact that this will increase the generalization ability of the model, it will also solve the problem of large dimensionality of data - because interests will not be described by a vector of goods, but by a significantly smaller vector of preferences.

Such approaches are also called spectral decomposition or high-pass filtering (since we remove noise and leave a useful signal). There are many different matrix decompositions in algebra, and one of the most commonly used is called singular value decomposition (SVD).

The SVD method was used in the late 80s to select pages that were similar in meaning, but not in content, and then began to be used in recommendations tasks. The method is based on the decomposition of the initial matrix of ratings into a product of 3 matrices:

$$R = U \times D \times S,$$

where (k,m)=(k,r)∗(r,r)∗(r,m) are the sizes of the matrices and r - decomposition rank - a parameter characterizing the degree of decomposition detail.

Applying this decomposition to our preference matrix, we obtain two matrixes of factors (abbreviated descriptions): U – a compact description of user preferences, S a compact description of product features.

It is important that with this approach we do not know which characteristics correspond to the factors in the reduced descriptions, for us they are encoded by some numbers. Therefore, SVD is an uninterrupted model.

In order to get an approximation of the preference matrix, it suffices to multiply the matrix of factors. Having done this, we obtain a rating score for all client-product pairs.

The general family of such algorithms is called NMF (non- negative matrix factorization). As a rule, the calculation of such expansions is very laborious, therefore, in practice, they often resort to their approximate iterative variants.

ALS (alternating least squares) is a popular iterative algorithm for decomposing a preference matrix into a product of 2 matrices: user factors (U) and product factors (I). It works on the principle of minimizing the standard error of the ratings. Optimization takes place alternately, first by user factors, then by product factors. Also, to circumvent retraining, regularization coefficients are added to the standard error.

If we supplement the preference matrix with a new dimension containing information about the user or the product, then we will be able to expand not the preference matrix, but the tensor. Thus, we will use more available information and possibly get a more accurate model.

F. Hybrid solutions

In practice, only one approach is rarely used. As a rule, several algorithms are combined into one in order to achieve maximum effect.

The two main advantages of combining models are increased accuracy and the possibility of more flexible tuning to different groups of customers. The disadvantages are less interpretability and greater complexity of implementation and support.

Several combining strategies:

- Weighting - reading the weighted average forecast for several estimates.
- Stacking - predictions of individual model inputs of another (meta) classifier that learns to correctly weight intermediate estimates.
- Switching - applying different algorithms for different products / users.
- Mixing – recommendations are calculated on different algorithms, and then simply combined into one list.

For example, content-based recommender is used, and one of the features is - the result of collaborative filtering.

Feature weighted (linear) stacking:

$$P(u,i) = w_1 P_1(u,i) + w_2 P_2(u,i) + \ldots + w_n P_n(u,i)$$

Weights w_1, w_2 ... w_n are trained on the sample. As a rule, logistic regression is used for this.

Stacking in general:

$$P(u,i) = f_1(u,i) P_1(u,i) + f_2(u,i) P_2(u,i) + \ldots + f_n(u,i) P_n(u,i)$$

G. Other approaches

1) Association Rules

Associative rules are generally used in the analysis of product correlations (Market Basket Analysis) and look something like this: “if there is milk in the customer’s check, then in 80% of cases there will be bread”. That is, if we see that the client has already put milk in the basket, it’s time to remind about the bread.

This is not the same as analysis of purchases spaced in time, but if we consider the whole history as one big basket, then we can fully apply this principle here. This may be justified when, for example, we sell expensive one-time goods (credit, flight).

1.1) RBM (restricted Bolzman Machines)

Bounded Boltzmann machines are a relatively old approach based on stochastic recurrent neural networks. It is a latent variable model and in this it is similar to SVD decomposition. It also looks for the most compact description of user preferences, which is encoded using latent variables. The method was not developed to search for recommendations, but it was successfully used in the top Netflix Prize solutions and is still used in some tasks.

1.2) Autoencoders

It is based on the same principle of spectral decomposition, which is why such networks are also called denoising auto- encoders. The network first collapses the user data it knows about into a compact representation, trying to leave only meaningful information, and then restores the data to its original dimension. The result is a kind of averaged, noise-free template that can be used to evaluate interest in any product.

2) DSSM (deep sematic similiarity models)

It is one of the new approaches using the same principle, but here the role of latent variables is performed by the internal tensor descriptions of the input data (embeddings). Initially, the model was created for query matching with documents (as well as content-based recommendations), but it is easily transformed into the task of matching users and products (Figure 11).

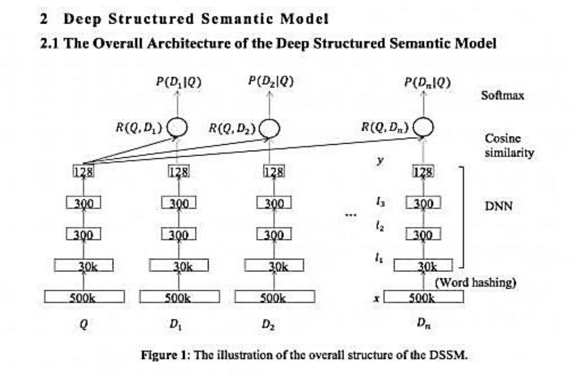

Figure 11. An overall architecture of DSSM

The variety of deep network architectures is unlimited, which is why Deep Learning provides a truly wide field of experimentation for recommender systems.

Experimental data

Online E-commerce websites like Amazon, AliExpress use various recommendation models to make different offers to users. Amazon right now uses a collaborative item-to-item filtering which grows to enormous datasets and delivers great high-quality recommendations progressively. This kind of filtering compares the purchased and valued items of each user to similar items then joins those corresponding items into a user recommendation list (Figure 12).

Figure 12. An example of recommendation in Amazon

In this article, an e-commerce product was taken as experimental data, based on 4 variables and more than 7 million items. Therefore, this article attempts to develop a recommendation model for Amazon's electronics products. Amazon is one of the world's largest e-commerce companies. They market millions of items around the world every day, adding multiple thousands to their range of products. It is very important that we continue analyzing the effectiveness of our products. However, most identical products are regulated differently, due to the varied digital infrastructure. Thus, product quality analysis primarily limits the ability to group related products in a precise manner. In order to develop the model, we will first use various types of recommendations systems, including popularity based systems, content based systems and collaborative filtering. We are flooded with tons of information in this contemporary world and that data yield the valuable knowledge. But customers can not obtain the information they are interested in from that data. Recommended systems have been implemented to help the client to figure out product details. A recommender system generates a correlation between the user and objects and employs the user / item commonality for making recommendations.

Table 1. Description of features in the dataset

Feature	**Type**	**Description**
userId	object	Every user is identified with a unique id
productId	object	Every product is identified with a unique id
Rating	float	Rating of the corresponding product by the corresponding user
timestamp	integer	Time of the rating

Table 1 illustrates that, each variables has its own specific type. The shape of the data: (7824482, 4). There are no missing values.

The total number of unique ratings is 7824482, whereas the total number of users is 4201696 and the total number of products is 476002.

As we can see in Figure 8, there is no equal distribution between ratings, rating 5.0 has been given by most users, whereas rating 2.0 has been given by less than half million, and it is the lowest one, comparing to others. The mean of the rating in the dataset exceeds 4.0. We divided our dataset into two parts, 70% of the dataset is training and 30% is the test dataset.

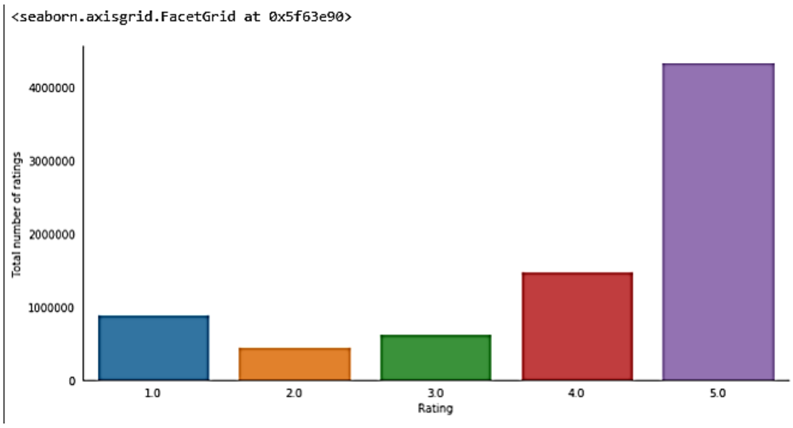

Figure 13. Distribution of ratings in the dataset

Note, the data points have been described by their features already; we are directly within the setting of the feature space. We will give some statics on this data, and it will be our initial analysis. Since, we have 4 features, we present only some relevant value to give an idea of each statistical collection.

Table 2. Statistics of the rating value

	Mean	Count	Standard deviation
Rating	3.972	1048576	1.399

Proposed methodology

The exponential rise in the amount accessible of digital information and the number of Internet users has generated a possible information overload problem that impedes rapid response to points of interest on the Internet. There were no problems related to the prioritization and personalization of data (where a program correlates the available content to the customer's desires and priorities).

The scheme for solving these problems is as follows.

1. Research analysis or the information collection phase. At this stage, we performed a one-dimensional and two- dimensional analysis of data, processing emissions, and missing values. The missing values were replaced by averages. In this project, there are no missing values. Also this step gathers accurate user data to produce a client’s profile page or model for predictive tasks along with the customer's rating, habits or content based on access resources. So the customer profile defines a basic user model. The effectiveness of every recommendation system is heavily dependent upon its ability to operate the current interests of users. Reliable models are important to get adequate and effective recommendations from any predictive techniques.

2. The learning phase. It implements a learning algorithm to sort and manipulate the features of the customer from the feedback obtained during the process of information collection. This effectively turns off all models except the one that fits best.

3. The prediction/recommendation phase. It suggests or forecasts what sort of products the user may choose. This could be achieved either through an assessment of the dataset obtained during the process of information collection which may be based on memory or model, or on the customer's experienced data (Figure14).

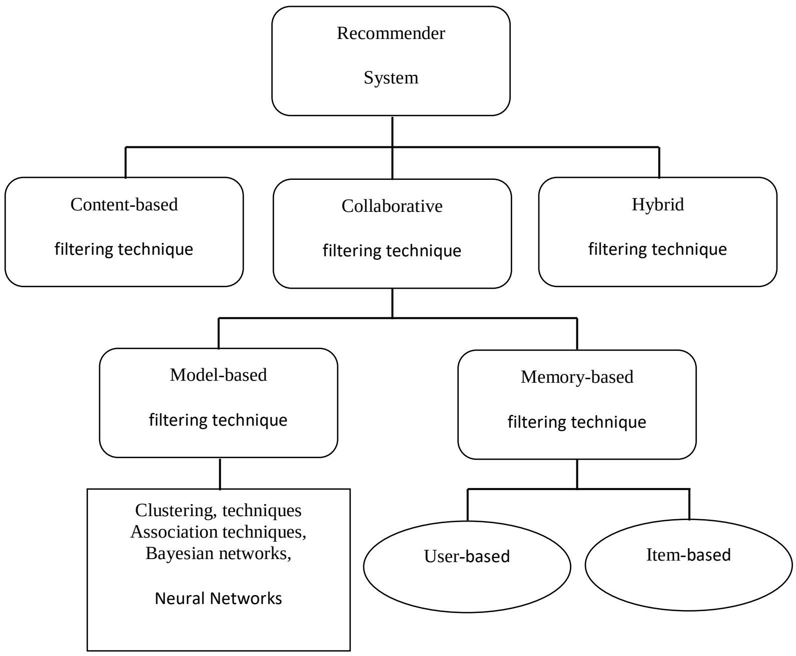

Figure 14. Recommendation techniques

A. Popularity Based Recommendation

The recommendation system based on popularity functions as a pattern. It uses the products that are currently in trend. For instance it indicates, whether any item that every new customer normally purchases is likely to be recommended to the customer who has just registered.

The new data frame includes customers who have given 50 ratings or more (Figure 15).

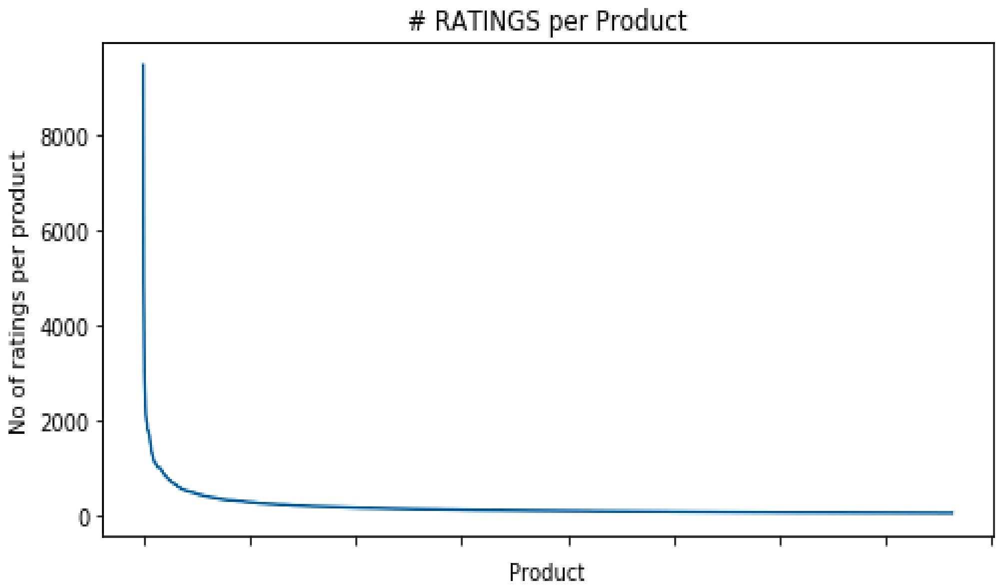

Figure 15. Number of ratings per product

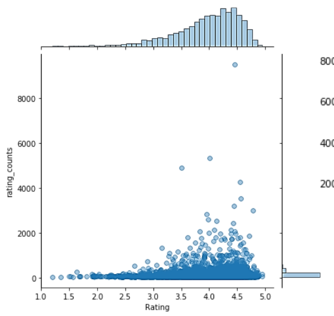

Figure 16. Rating vs rating counts

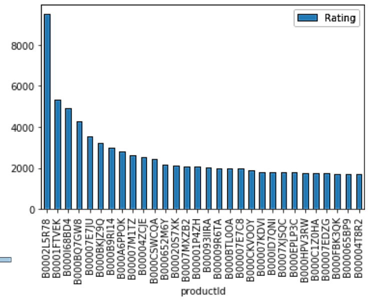

Figure 17. Final sorting of popular products by rating

Collaborative filtering (Item-Item recommendation)

For recommender systems collaborative filtering is widely implemented. Such approaches are intended to replace the missing elements of a matrix association of customer- items. We will be using the method of collaborative filtering. This uses historical item ratings by around-minded individuals to determine how others will classify the item in question. Collaborative filtering has two subgroups that are commonly referred to as memory-based and model-based approaches. After splitting the dataset into train and test (70/30 correspondingly), where the random state is equal to 10, we perform KNNWithMeans algorithm, taking into account the mean ratings of each user, where the parameter k is 5. Next, after computing the Pearson baseline similarity matrix and fitting the train set, we can see the result of our trained model against the test set. The RMSE-based accuracy measure is equal to 1.3436.

B. Model-based collaborative filtering system

These methods are mainly based on the techniques of machine learning and data mining. The aim is to train models so that they can draw conclusions. For instance, we might use current user-item relationships to train a model to predict the top-five products that a customer could perhaps like the most. One benefit of these approaches is that they can suggest a greater number of products to a wider range of users in comparison to other techniques such as memory-based approach. We have such a wide range, even though dealing with big, sparse matrixes (Figure 18).

```
ratings_matrix = new_df1.pivot_table(values='Rating', index='userId', colum
ns='productId', fill_value=0)
ratings_matrix.head()
```

productId	0972683275	1400501466	1400501520	1400501776	14005326
userId					
A01852072Z7B68UHLI5UG	0	0	0	0	0
A0266076X6KPZ6CCHGVS	0	0	0	0	0
A0293130VTX2ZXA70JQS	5	0	0	0	0
A030530627MK66BD8V4LN	4	0	0	0	0
A0571176384K8RBNKGF8O	0	0	0	0	0

Figure 18. Results of sparse matrix

As predicted, the utility matrix above is sparse, and the unknown values are marked as 0. The shape of the matrix is (9832, 76). After transposing this matrix, the shape has changed to (76, 9832). If we compare these two matrices we can see the unique products in this subset of data. The next stage is decomposing the matrix using truncated SVD (singular value decomposition) and then building a correlation matrix on the decomposed matrix. If we choose one product item, correlation for all items with the item purchased by this customer is based on the items rated by other customers' people who bought the same product. Recommending top 25 highly correlated products in sequence - removes the item already bought by the customer.

Conclusion

Recommendation systems create new opportunities for online retrieval of useful information. This also aims to relieve the big data challenges, which is a very frequent occurrence with knowledge extraction systems, and allows users to access goods and services that are not easily and quickly provided to the system users.

This article describes the two traditional methods of recommendation in terms of their advantages and weaknesses using various types of synthesis methods to enhance their efficiency. Throughout this study the authors have performed research assessments of different stages and methods within the recommender systems. The study has revealed that collaborative the user and user filtering has a higher performance than the other methods as it yields more accurate results than the item-item filtering. However it must be admitted that each method has benefits and drawbacks. Different learning algorithms have been used to develop recommendation models and assessment metrics for evaluating the consistency and efficiency of the recommendation algorithms. The article describes the process of processing the client-received data, in the recommender system. For greater precision the number of iterations will be increased.

The task of creating recommendations is quite easy, we compile a preference matrix with defined figures, as it turns out, we supplement these forecasts with consumer and product details and try to fill in the unknown values. Notwithstanding the simplicity of the formulation, hundreds of articles have been published which explain basically new ways for solving it. For starters, this is attributed to an increase in the selection of data that can be included in the model, as well as an increase in the importance of implicit ratings. Second, the rise of deep learning and the advent of modern neural network architectures adds to the models' difficulty. These findings inspire the authors to develop an action plan for further research in this area.

References

[1] Rashid AM, Albert I, Cosley D, Lam SK, McNee SM, Konstan JA et al. Getting to know you: learning new user preferences in recommender systems. In: Proceedings of the international conference on intelligent user interfaces; 2002. p. 127–34.

[2] Schafer JB, Konstan J, Riedl J. Recommender system in e-commerce. In: Proceedings of the 1st ACM conference on electronic commerce; 1999. p. 158–66.

[3] P. Resnick, H.R. VarianRecommender system's Commun ACM, 40 (3) (1997), pp. 56-58, 10.1145/245108.24512

[4] Jannach, D., Zanker, M., Felfernig, A., Friedrich, G. (2010). Recommender systems – an introduction. Cambridge: Cambridge University Press.

[5] Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J. (1997). Grouplens: Applying collaborative filtering to usenet news full text. Commission of the ACM, 40(3), 77–87.

[6] Pazzani, M., & Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. Machine Learning, 27, 313–331.

[7] Felfernig, A., Friedrich, G., Jannach, D., Zanker, M. (2015). Constraint- based recommender systems. In Recommender systems handbook, pp. 161–

190. Springer.

[8] Felfernig, A., Boratto, L., Stettinger, M., Tkalčič, M. (2018). Group recommender systems: an introduction. Springer.

[9] Masthoff, J. (2011). Group recommender systems, Recommender Systems Handbook, pp. 677–702.

[10] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. UMUAI Journal, 12(4), 331–370.

[11] A.M. Acilar, A. ArslanA collaborative filtering method based on Artificial Immune Network

Exp Syst Appl, 36 (4) (2009), pp. 8324-8332

[12] L.S. Chen, F.H. Hsu, M.C. Chen, Y.C. HsuDeveloping recommender systems with the consideration of product profitability for sellers Int J Inform Sci, 178 (4) (2008), pp. 1032-1048

[13] M. Jalali, N. Mustapha, M. Sulaiman, A. MamayWEBPUM: a web- based recommendation system to predict user future movement

Exp Syst Applicat, 37 (9) (2010), pp. 6201-6212

[14] S.H. Min, I. HanDetection of the customer time-variant pattern for improving recommender system Exp Syst Applicat, 37 (4) (2010), pp. 2911-2922
[15] O. Celma, X. SerraFOAFing the Music: bridging the semantic gap in music recommendation Web Semant: Sci Serv Agents World Wide Web, 16 (4) (2008), pp. 250-256
[16] G. Adomavicius, A. TuzhilinToward the next generation of recommender system. A survey of the state-of-the-art and possible extensions IEEE Trans Knowl Data Eng, 17 (6) (2005), pp. 734-749
[17] G. Murat, G.O. SuleCombination of web page recommender systems Exp Syst Applicat, 37 (4) (2010), pp. 2911-2922
[18] Mobasher B. Recommender systems. Kunstliche Intelligenz. Special Issue on Web Mining, BottcherIT Verlag, Bremen, Germany, vol. 3; 2007. p. 41–3.
[19] M.Y.H. Al-Shamri, K.K. BharadwayFuzzy-genetic approach to recommender systems based on a novel hybrid user model
[20] Expert Syst Appl, 35 (3) (2008), pp. 1386-1399

ИСПОЛЬЗОВАНИЕ МАШИННОГО ОБУЧЕНИЯ В РЕКОМЕНДУЕМОЙ СИСТЕМЕ В ЭЛЕКТРОННОЙ КОММЕРЦИИ

Е. АХМЕР
PhD докторант кафедры компьютерной инженерии Международного университета информационных технологии

Г.У. БЕКТЕМЫСОВА
Ассоц.профессор кафедры компьютерной инженерии Международного университета информационных технологии s

Кафедра вычислительной техники
Международный университет информационных технологий, Алматы, Казахстан
Электронная почта: g.bektemisova@iitu.edu.kz

Аннотация. Метод фильтрации незаменим в среде, перегруженными данными. Рекомендуемые системы сделали огромный шаг к этой цели, ускорив работу с клиентами через Интернет. Большинство современных примеров искусственного маркетингового интеллекта известны как контролируемое обучение, которое варьируется от предложения персонализированных конкретных продуктов с определением наиболее ценных маркетинговых стратегий до прогнозирования скорости оттока клиентов или ценности жизни клиентов и создания положительной клиентской базы. Как правило, различные типы хранимой информации используются для настройки различных параметров или результатов поиска, демонстрации наиболее целевой рекламы на главной странице и т. д. Рекомендуемые системы получают прибыль, используя предложения для увеличения продаж. Любая другая система может использовать разные данные из нескольких источников для оценки моделей использования и обнаружения схожих тенденций, которые позволяют прогнозировать будущие покупки или предпочтения клиентов. Он предсказывает интересные закономерности и предоставляет рекомендации на основе модели интересов клиентов. Кажется, с одной стороны, традиционная система рекомендаций, которая предлагает товары на основе различных критериев потребителей или продуктов, таких как цена продукта, информация о пользователе и т. д., но, с другой стороны, мы также рекомендуем системы, включающие глубокое обучение. методов, даже если они еще недостаточно изучены. В этой статье обсуждаются различные процессы, связанные с внедрением рекомендательных систем, и многочисленные рекомендательные подходы, а также анализ этих методов, которые могут использоваться разными учеными в нескольких статьях. Внедрение метода совместной фильтрации и методов фильтрации на основе контента в значительной степени бессмысленно, поскольку большинство интернет-магазинов уже используют гибридные механизмы, которые оказались более эффективными. В нашем исследовании мы также включили преимущества и недостатки каждого подхода. В этой статье также представлены многочисленные трудности и проблемы, с которыми сталкиваются рекомендатели в алгоритмах своих прикладных систем. В этой статье мы сначала представляем несколько наиболее известных типов рекомендуемых систем и концентрируемся на одной части рекомендаций для электронной коммерции, а затем проводим их количественное сравнение. Рекомендательные системы сделали огромный шаг к этой цели, значительно улучшив пользовательский опыт в онлайн-среде.

Ключевые слова*:* системы рекомендаций, электронная коммерция, фильтрация на основе контента, совместная фильтрация, фильтрация на основе пользователей.