Investigation of Recommender System Failure Scenarios

Yewei SONG
School of Computer Science
University of Birmingham
Birmingham, UK
yxs987@student.bham.ac.uk

***Abstract*—This investigation is about fail scenarios in recommender system. And what methods and techniques are used to solve these failure scenarios. This article mainly studies how to improve these failure scenarios from five aspects: personal adaptation, recommendation quality, privacy protection, ethical principles, and deep models.**

***Index Terms*—Recommender System, Human computer interaction**

I. Introduction

Nowadays, anyone who is an Internet user can feel the existence of a recommendation system. On Amazon, the system will recommend products to us based on shopping history; on Youtube, autoplay will continue to play related videos; on the music app, the system recommends playlists to users. But are these recommendations necessarily good? Duplicate and useless recommendations, information cocoon issues [24], privacy disclosure issues, personally unrelated recommendations, cold start issues [34]. These issues have plagued recommendation systems. This article focuses on exploring the existence of these problems and possible solutions.

II. Previous development

A. Why we need Recommender System?

Modern society is facing serious information overload, and the meaning of information overload means that the total amount of information in the entire society exceeds the processing capacity of the individual, and the individual cannot accept all the information [39]. This means finding the information you need like a needle in haystack. To solve the difficulty of finding information, people first used the Internet yellow pages in the 1990s [14]. But with the further increase of information, the yellow pages became more and more difficult to use. Founded in 1998 by Larry Page and Sergey Brin, Google Corporation designs and manages the Internet search engine "Google." [68]. It changed the Internet in the early 21st century, and people could use keywords to get the information they needed.

But all of these solutions are based on a model in which users actively obtain information. The problem of how service providers proactively provide information to customers is not resolved until the appearance of a recommender system. Prior to the recommendation system, it was difficult for Internet companies to target their services and products to targeted customers. And customers need to repeatedly adjust the search keywords for their own situation to reach the target page.

The user does not need to actively tell the system what kind of data the user is interested in, nor does it need to provide keywords. The system provides the information the user needs based on the user's historical behavior and preference data.This is also a common situation, that is, the user does not know what they need.A good recommender system can tap the needs of users and even create orders for goods that were otherwise impossible [25].

B. What Recommender System is ?

Resnick and Varian first define of the recommendation system in 1997 [55]: *"It uses e-commerce websites to provide customers with product information and advice, help users decide what products to buy, and simulate sales staff to help customers complete Purchase process."* The task of the recommendation system is to contact users and information. On the one hand, the recommendation system can actively provide the information of the supplier to the customer, on the other hand, the system can also help users find valuable information. This process also provides value to both parties and avoids problems that are difficult to find in the ocean of information. ocean. [71].

In 1985, scholars such as David K. Gifford published An architecture for large scale information systems, which laid the foundation for research on recommendation systems [22]. In 1988, Stephen Pollock described the ISCREEN system for filtering text messages [51]. ISCREEN includes a high-level interface for defining rules, a component for displaying text messages on screen, and a conflict detection component for checking for inconsistencies. This is an early filtering system. In 1990, Ernst Lutz and other scholars proposed MAFIA. At that time, the message filtering system required that the messages to be processed be at least semi-structured [38]. The MAFIA system (MAil-FIlter-Agent) overcomes this limitation by providing an automatic document classification component and automatically recognizes it. Related concepts for weakly structured documents. In 1992, David Goldberg and other scholars proposed Tapestry, the first collaborative filtering system [23]. This is a mail system developed at the Xerox Palo Alto Research Center. Tapestry's motivation comes from the increasing use of e-mail, which has caused users to be overwhelmed by a large number of incoming documents. One way to handle a lot of mail is to provide mailing lists so that users can subscribe to only the lists they are interested in. A better solution is to let the user specify a filter to scan all lists and select documents of interest, regardless of which list the message is in. This is the design philosophy of Tapestry. A basic principle of Tapestry's work is that it can be filtered more effectively by involving humans in the filtering process [2] [23].

The recommendation system to become a relatively independent research direction is generally considered to be the GroupLens system introduced by the GroupLens research group of the University of Minnesota in 1994 [32]. The most outstanding contribution of this system lies in the design and implementation of a collaborative filtering-based recommendation system for the first time. In the subsequent design of the recommendation system, all latecomers cannot avoid the use of collaborative filtering technology. The team's design ideas also extend to other product settings, such as books and movies. The research group also designed a recommendation system based on book and movie information, namely BookLens and MovieLens. [28] [3].

In 1998, John S. Breese and other scholars evaluated the performance of user-based collaborative filtering systems. The algorithms used included techniques based on correlation coefficients, vector-based similarity calculation, and statistical Bayesian methods. They use two basic evaluation indicators. The first indicator measures the accuracy of a single set of predictions, using the average absolute error as an indicator. The second indicator measures the effectiveness of the recommended item ranking list, with users in an ordered list [27]. See the estimated value of the recommended probability as an indicator. In 2001, Badrul Sarwar and his team compared different item-based recommendation generation algorithms, mainly from the technology for calculating similarity of items and the recommendation generation that should be used subsequently Compare the two aspects of the model. They use the k-nearest neighbor method as a benchmark. The comparison results show that the item-based algorithm performs better than the user-based algorithm, and at the same time, it can provide better recommendation results than the best user-based algorithm [59].

Thomas Hofmann (1999) proposed pLSA, which relies on a hybrid decomposition method. He then conducted a series of empirical studies and discussed the application of pLSA in automatic document indexing. His empirical results show that pLSA has improved significantly compared to LSA [29]. At the end of the same year, Will Glaser and Tim Westergren proposed the idea of the Music Genome Project. In January 2000, they teamed up with Jon Kraft to create Savage Beast Technologies to bring their ideas to implements [8]. This is a typical early business practice of recommendation systems. Since then, the implementation of recommendation system technology has become faster and faster. For example, Last.fm began using recommendation algorithms in 2002 to recommend music that users might like [26]. The first company to use collaborative filtering technology in a commercial system should be Netflix. They have achieved good results, and many companies have subsequently adopted this technology. For example, on the Amazon website, they use a recommendation algorithm to personalize the online store for each customer [62]. In their published paper, they proposed item-to-item collaborative filtering, and worked with common methods-traditional collaborative filtering, cluster models, and search-based methods.a comparison was made [37]. The online calculation scale of their algorithm has nothing to do with the number of customers and the number of items in the product catalog, and can generate recommendations in real time, or it can be extended to massive data sets to generate high-quality recommendations. Netflix has also been continuously researching in this field. In the Netflix Prize competition, Yehuda Koren considered that because customer preferences for products have changed over time, modeling time dynamics. For designing recommendation systems or general customer preference models It is essential to propose a dynamic recommendation system [33]. With the increasing importance of recommendation systems, the ACM RecSys conference was founded in 2007 and has become the best specialized conference in the field of recommendation systems [61].

C. How Recommender System work ?

There are two common working methods of recommendation systems, collaborative filtering and content-based recommendation and hybrid system [6].

The collaborative filtering method is to find the relevance of the item or content based on its metadata, and then recommend similar items to the user based on the user's previous preference records [60]. Collaborative filtering recommendations can be divided into the following three categories

- **User-based Collaborative Filtering Recommendation**:The user-based collaborative filtering algorithm finds the user's likes (such as product purchases, favorites, content reviews, or shares) through the user's historical behavior data, and measures and scores these preferences. Calculate the relationship between users based on their attitudes and preferences for the same product or content. Make product recommendations among users who have the same preferences [75]. One case is that if both A and B have purchased the same items, and both have given full marks after purchase, we may be able to attribute two people to the same type of user. If A later praises a new product, we will push it to B [45].
- **Item-based Collaborative Filtering Recommendation**The biggest difference between Item-based collaborative filtering and user-based collaborative filtering is that the former focuses on finding similarities between products. [74]. If we find that people who give a good score for product A will give a high score for product B, then we consider that product A and B have a high correlation. [59]. For example, if you bought an android

book, then the sales website is likely to recommend you interface design or java books for you.

- **Model-based Collaborative Filtering Recommendation**:Model-based collaborative filtering attempts to quantify how a user would like items they haven't encountered before [30]. Model-based collaborative filtering attempts to use machine learning to extract certain special discounts on goods or people, such as a vector. Then use machine learning methods to predict the match between users and products. Common model used in collaborative filtering are Bayesian networks, singular value decomposition, and implicit probabilistic semantic analysis.

The content-based recommendation algorithm is mainly based on the user's preference history, such as purchase history or watching history.The first step is to extract some content for each item to represent it. The second step is to use the feature data of an item that the user liked (and did not like) in the past to learn the user's favorite profile. Finally, by comparing the characteristics of the user profile and candidate items obtained in the previous step, a set of items with the highest correlation are recommended for this user [50].

D. How it influence Human computer interaction ?

Algorithm researchers believe that if the system recommends the content to the user, the user should explicitly accept the recommendation. Obviously, this extremely simplifies the recommendation problem. Indeed it is hard to pass the recommendation results to the user [77].

The effectiveness of a recommendation system depends on many factors, not just the quality of the prediction algorithm. In fact, the recommendation system needs to give users the opportunity to understand the recommended items, including text, voice, pictures, videos, and so on. Choose how to display product-based attributes, such as product images, video screenshots, and game CGs. [58]. The process of presenting recommended items to users is highly dependent on human-computer interaction design. Good human-computer interaction design can effectively improve the success rate of recommendations. [56].From the user's point of view, the recommendation system must be credible and transparent, it must inform the user of the reason and logic of the recommendation, it must guide the user to use unfamiliar new items, and it must provide detailed information about recommended items.

In Francesco Ricci's Recommender Systems Handbook [57], he makes three important points. First, trust is important, and one of the ways to build trust is to explain the reason for the recommendation. For example, "This item is recommended because you say you own item X" or "The recommendations for these items are based on items you have recently viewed." The main advantage of this type of system is that users can know what the nature of the recommendation system is, such as how these recommendation results are determined, so that users will trust these recommendation results more.

Another strict limitation of recommendation system algorithms is that these algorithms are designed to collect input data at one time, and end the whole process once the recommendation results are returned. In many cases, this model will fail because users are not yet fully aware of their preferences unless they interact with the system to some degree and have a general understanding of the range of choices. Or browse through more options before you are convinced that some of them may be suitable for you. With good interpersonal interaction design, users will be happy to provide feedback, and feedback can effectively improve the quality and accuracy of recommendations. So users get higher quality recommendations [56].

The third point is visualization. How the system displays the recommendation results is a key factor affecting the user's acceptance of the recommendation results and the recommendation system.

E. Most popular Recommender System

Among the most successful recommendation systems in the industry, Google(YouTube), Facebook and Amazon are the most representative.

Currently, the data set faced by Facebook's recommendation system includes more than 1 billion records, and it has 100 times more data than Netflix Prize2 [48]. To solve this big scale of data, Facebook has designed a new recommendation system, Deep Learning Recommendation Model.On the one hand, neighborhood methods are used to group users and products and matrix decomposition is used to describe latent factors of users and products. On the other hand, deep networks are used to classify or predict the event probability of a given data. In order to deal with category features, embeddings are generally used to transform one-hot or multi-hot vectors into dense respresentations of abstract space. DLRM combines the above two angles. The model uses embeddings to handle sparse features, MLP handles dense features, and then uses statistical techniques to display the feature intersections. Finally, use another MLP to process the crossover features and get the probability of the event [44].

Amazon has the "king of recommendation system" in the industry, and Amazon claims that 35% of its sales are related to the recommendation system. But Amazon's system is more based on the principle of human-computer interaction. For example, explain the reason for recommendation, use product ratings to weight recommendations, and ultimately build user trust in Amazon's recommendation system. Specific recommendations are also classified recommendations based on purchase history and user portraits [64]. For example, Figure 1 is the purchase history recommendation, and Figure 2 is the user interest category recommendation.

III. Defects of existing system

A. Duplicate or useless recommendation

A most typical scenario is that the user searches for and purchases a product, but in the next few days, the recommendation system tirelessly pushes the same or similar products to the user. In this scenario, the user does not need to repeat the purchase at all, and the recommendation behavior can be considered invalid. Another common scenario is when the user simply understands some information, such as searching

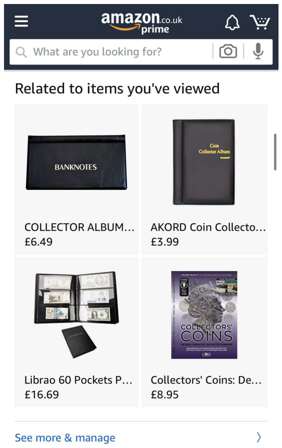

Fig. 1. Recent viewed recommend

Fig. 2. User model recommend

on Google for "how to use a printer". The recommendation system is likely to think that you need to buy a printer, and in fact users who search for printers are likely to already own or not need a printer.
This problem most often occurs in content-based recommendation systems. Because the system does not know the specific actions after the user's search or access is complete, the user's interests are only inferred based on the user's behavior.
Another common scenario is that after a user understands a certain content X, the user is not interested in the content X. But the recommendation system recommends content X based on the user's behavior history.Such false recommendation causes users to have a distrust and aversion to the recommendation system. At the same time, repeated recommendations also waste the opportunity to recommend related products or content.

B. Not applicable to the individual

User-based Collaborative Filtering Recommendation is usually an average analysis based on a large number of similar user data. User-based Collaborative Filtering Recommendation is usually an average analysis based on a large number of similar user data. In other words, push the preference products of similar users, but do not consider individual differences [1]. Most evaluation indicators do not take into account the differences of individual users: open-minded users may prefer highly novel or diverse recommendations, while conservative users may not require as much novelty or diversity.
Another common misfit is "over-sensitivity" for occasional situations. For example, a user is suddenly invited to a wedding one day, and then the user purchases some souvenirs and a red tie. But the recommendation system is likely to think that the user prefers red ties or purchases small gifts.

C. Privacy protection

Although the recommendation system facilitates users to obtain information, the historical data set, prediction model and recommendation results of the recommendation system are all closely related to the privacy of users.The more effective the recommendation system, the more dimensions of user data are required. From visit history and purchase history to location information and social network content. There are two main problems in the process of data collection and utilization.
According to a 2018 research paper published in Engineering [69], the data "undesirably discloses the users' personal interests to the recommender." Data collected by commercial companies may be sold to third-party analysis companies without the user's permission. Another problem is that platform users' personal data can be leaked, and Facebook (and other platforms) have happened several times [10] [15].
Another common problem is excessive information collection. In order to build a more effective user model, Internet companies usually collect user information as much as possible, including location, call history, contacts, and photo albums. In some cases, certain user privacy can even be obtained through the bypass, such as judging the user's travel trajectory by using EXIF information in the album [46]. Or use the address book to build a relationship network, and then use the call record to judge the relationship. Such behaviors can cause strong resentment from users in some cases, especially without explicit permission.

D. Ethical issues

"One ethical issue in recommendation systems is that they are created to be addictive." -Haley DeLeon [43]
For example, YouTube's autoplay function, which always plays your current video or history-related content, tempting people to watch continuously. Facebook founding member Sean Parker acknowledges in an interview with Axios when these big companies are using human psychology to create addictive products.
"The thought process that went into building these applications, Facebook being the first of them, ... was all about: 'How do we consume as much of your time and conscious attention as possible.' [35]"
But so far, no one has proposed separate information addiction as a mental illness.
The second ethical issue is Extreme Content. In order to gain the user's attention, the content may gradually be polarized. When the Google can obtain advertising profits from users

who continuously watch radical content, the reward mechanism of the recommendation algorithm has also begun to promote users to polarize As Zeynep Tufekci puts it in The New York Times,

"YouTube leads viewers down a rabbit hole of extremism, while Google racks up the ad sales. [66]"

Although users may have begun to watch videos with a neutral point of view, YouTube 's autoplay system still pushes users to watch supremacist and conspiracy theory videos. Even for non-political topics, such as environmental protection, the system will finally recommend videos to users with extreme environmentalism

IV. Possible improvement

A. Personal fitting for individuals

In view of the differences between statistical models and personal characteristics, the recommendation system should establish localized and customized personal models for individual users.Most generalized recommendation systems cannot effectively target Heavy-tailed Distribution. At the same time, excessive optimization of general-purpose systems can cause serious diminishing marginal benefits.

Users' personal preferences can be divided into long-term preferences and short-term preferences. Long-term preferences often refer to the interests of users [73]. For example, she is a fan of Taylor Swift, so she will be interested in Taylor Swift songs and concert tickets for a long time. Short-term preferences refer to users' immediate interests in the current environment. For example, in the past week, users prefer to listen to popular songs on spotify, then the recommendation system should also capture the user's interest, or if the user plans to move in the next month, then the recommendation system can appropriately push some moving company ads. Improved collaborative filtering using neural networks can deal with long-term and short-term preferences to a certain extent, but it is still necessary to consider the period of preference decay.

The author proposes a way to improve the recommendation system for e-commerce systems. Existing e-commerce systems often recommend duplicate products to users based on purchase access records. If we add a life cycle for each consumable product, for example, the life cycle of a shampoo is one bottle per month, and the life of a hair dryer is several years. When the user purchases a product, we give the product a large negative weight coefficient and restore his weight according to the logarithmic curve until the replacement cycle is approaching. Under this design, the user will find that the recommendation system will give effective recommendations whenever the consumables purchased by him/her are running out, which can increase the user's trust in the recommendation system. Consumable cycles can also be corrected automatically based on purchase records.

Another feasible recommendation idea is to recommend related products as much as possible instead of the same products. When establishing the item relationship network model, we target some products with special relationships, such as printers and ink cartridges (consumables model), baking trays and tin foil (cooperative model), luggage and storage bags (optimized model). In the case where the user purchases the product A, a category B having a special relevance is recommended. When the user finds that the recommendation system recommends the necessities that the user has forgotten to purchase, it also increases the trust in the recommendation system.

Localization modeling is also a popular way to build recommendation models. Collect daily behaviors of users, such as actively collecting working hours and cycles, and users' anniversaries. Finally we can get the user's time-preference model. This is to avoid recommending vacation plans at inappropriate times such as during work hours, or recommending a single working day meal on a wedding anniversary.But this process should pay attention to privacy protection issues very careful, we will describe in later chapters.

B. Advertisements quality

We can improve the quality of the recommendation system from several main evaluation dimensions. Some indicators can be calculated quantitatively, and some can only be described qualitatively.

User satisfaction is an important indicator for evaluating recommendation systems. It cannot be calculated offline, but can only be obtained through user surveys or online experiments. When using a questionnaire survey, the user's feelings in all aspects need to be taken into account before the user can give an accurate answer to the question. In online systems, user satisfaction is obtained by counting user behavior. The conversion purchase rate brought by a recommendation can reflect the user's satisfaction with the recommended content, because it means that the recommendation system has achieved its purpose. The evaluation after the user purchases the product can also be used as a source of satisfaction data [36]. In non-shopping situations, we can measure user satisfaction with indicators such as user clickthrough rate, dwell time, and conversion rate.

Predict accuracy is an important indicator for offline evaluation. In order to calculate the accuracy rate, a part of the user behavior records are used as a test set, and then other user historical data is used to train a prediction model that predicts the user's interest. Then test the difference between the performance of the model on the test set and the actual performance of users in the test set. Because offline recommendation algorithms have different research directions and different accuracy indicators, according to the research direction, they can be divided into: prediction score accuracy and TopN recommendation.

The rating function is one of the key functions of many shopping sites and review sites. A user's historical rating data can reflect user preferences. [42]. The behavior of predicting a user's rating of an item is called rating prediction. The prediction accuracy of the score prediction is generally calculated by the root mean square error (RMSE) and the average absolute error (MAE) [9] [52]. For a user u and item i in the test set, let r_{ui}be the actual score of user u on item i and $\hat{r}_{ui}$ be the predicted score given by the recommendation algorithm, then RMSE is:

$$RMSE = \frac{\sqrt{\sum_{u,i \in T} (r_{ui} - \hat{r}_{ui})^2}}{|T|}$$

MAE uses absolute values to calculate prediction errors:

$$MAE = \frac{\sum_{u,i \in T} |r_{ui} - \hat{r}_{ui}|}{|T|}$$

Websites don't usually offer only a single recommendation, but a scrolling sorted list of N items. This type of recommendation is called TopN recommendation [13]. The prediction accuracy rate recommended by TopN is generally measured by 2 indicators:

$$\begin{aligned} Precision &= \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|} \\ Recall &= \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \end{aligned} \tag{1}$$

Coverage describes the ability of a recommendation system to resolve long-tail effects. The ability to discover items that are not often used and recommended. Usually defined as the ratio of recommended projects to total projects [12]. Assuming that the user set of the system is U, the recommendation system recommends to each user an item list R (u) of length N, and the coverage formula is [72]:

$$Coverage = \frac{|U_{u \in U} R(u)|}{|I|}$$

Coverage is an index that content providers care about [19]. A recommendation system with 100% coverage means recommend each item to at least one user. In addition to the proportion of recommended items, you can also better describe the ability of the mining system to mine long tails by studying the distribution of the number of times items appear in the recommendation list [49]. If the distribution is relatively flat, the coverage of the recommended system is high; if the distribution is steep, the coverage of the distributed system is low.

Diversity describes the dissimilarity between the items in the recommendation list.In order to satisfy a wide range of interests of users, the recommendation list needs to be able to cover areas of different interests of users, that is, it needs to be diverse. Assuming that s (i, j) defines the similarity between items i and j in the interval [0,1], then the diversity of the user u's recommendation list R (u) is defined [76] as follows:

$$Diversity = 1 - \frac{\sum_{i,j \in R(u), i \neq j} s(i,j)}{\frac{1}{2} |R(u)| (|R(u)| - 1))}$$

The overall diversity of the recommendation system can be defined as the average of the diversity of all user recommendation lists [47]:

$$Diversity = \frac{1}{|U|} \sum_{u \in U} Diversity(R(u))$$

Novelty is also one of the important indicators affecting user experience. It refers to the ability to recommend non-popular items to users. To evaluate novelty is to calculate the average popularity of the recommendation results, because the less popular items, the more likely they are to make users feel novel. This indicator mainly relies on user surveys and crawler analysis [7].

Serendipity is the hottest topic in the field of recommendation systems in recent years. If the recommend items is different from the user 's historical interests, but the user is satisfied, then it can be said that the recommendation result has a high degree of surprise, and the novelty of the recommendation depends only on whether the user has heard of the recommendation result [19].Recommendation results that surprise users are recommendations that are not similar to items they have liked in the user 's history, but that users find satisfactory. Then, to define the surprise degree, firstly define the similarity between the recommendation result and the user's favorite items in the history, and secondly define the user's satisfaction with the recommendation result [11].

C. Mask privacy data and lawful usage

Homomorphic encryptionis an encryption method that allows direct operation on ciphertext, and it has the same effect as operating on plaintext and then encrypting it.Homomorphic encryption can be used to transfer encrypted data to public cloud computing for processing without transmitting the original text [20]. In areas where privacy protection is highly regulated, new services can be enabled using homomorphic encryption by removing privacy barriers that prevent data sharingḞor example, financial product recommendation predictions may be difficult to apply due to financial data privacy issues, but if the recommendation system service provider can operate on full homomorphic encrypted data, these privacy issues will be reduced.

Differential privacy protection was defined as a new privacy protection model by Dwork in 2006.The task of differential privacy is to provide a mechanism or protocol about the probability distribution of the output, allowing users to modify the data to a certain extent, but without affecting the overall output, so that the attacker cannot know the information about the individual in the data set [16]. Differential privacy refers to the existence of two data sets X and X' that differ by at most one record and a privacy algorithm A. $Range(A)$ is the value range of A. If algorithm A on data sets D and D' When the output result $O(O \in Range(A))$ satisfies $Pr[A(X) = O] \leq e\epsilon \times Pr[A(X') = O]$, A satisfies the requirements of ϵ -differential privacy [17] [53]. The most important of differential privacy The method is implemented by adding noise to the data set. Common noise mechanisms include Laplace mechanism and exponential mechanism. Laplace mechanism is suitable for continuous data sets [63], while exponential mechanism is suitable for discrete data sets [31].

In 2009, McShery and others first introduced differential privacy to collaborative filtering systems. They used item-to-item covariance matrix differential privacy processing, and trained a predictive model using a standard test / training set approach. Their paper proves that using differential privacy in collaborative filtering does not seriously affect the accuracy of recommendations [41] [54].

Intelligent Anti-Tracking(ITP) is an initiative by Apple to protect the privacy of Safari users. The first version of the ITP was released in 2017 and is mainly for the use of third-party cookies [21]. In fact, Apple completely blocks third-party cookies.Although this approach helps protect user access records, it has caused serious problems for ad technology companies and marketing technology companies.

Do Not Track(DNT) is a browser function that can prevent users from being tracked by third-party websites that they have never visited. DNT does not use any means to filter or block tracking cookies. , "Do not track" browsers will add a "headers" to the HTTP data transmission, this header field will tell commercial server users do not want to be tracked [65].

With the successive introduction of **personal data protection laws** in various countries around the world, the European Union's *General Data Protection Regulation, 679/2016, GDPR* has been fully implemented on 25 May 2018.According to the GDPR, as long as any company collects user data, it must obtain the user's permission, and the user agreement must be written clearly and easily. Users can delete personal information authorized for corporate use and prevent previously authorized third parties from continuing to use the information. Enterprises also need to set up a full-time data protection person in charge and notify users within 72 hours of any possible data leak. [4]

D. Ethical

A new paper published by Google suggests updates to the YouTube platform algorithm. The researchers specifically raised a question they considered "hidden bias." It refers to the way the recommendation itself can affect the behavior of the user, making it difficult to tell whether you clicked on a video because it liked it, or because it was highly recommended [5]. As a result, over time, the recommendation system may push users further and further, preventing users from seeing the videos they really want to watch. To reduce this bias, researchers recommend tweaking the algorithm: Every time a user clicks on a video, it also considers the ranking of the video in the recommendation sidebar. When entering a machine learning algorithm, videos that are closer to the top of the sidebar are given less weight; videos that are ranked lower (requires user scrolling) are given more weight. When researchers tested these changes in real time on YouTube, they found that user engagement had increased significantly.

Another important measure to mitigate the Matthew effect is to conduct a satisfaction survey of the recommended content. Use satisfaction instead of total viewing time for performance evaluation [70]. The advantage of this is that you can evaluate both satisfaction and novelty and serendipity.At the same time, such changes can reduce user addiction caused by excessive pursuit of user retention.

There is another important function, maybe many pages have been provided, is the DISLIKE button. A well-designed dislike button should have a short feedback option with common reason categories, such as low quality content, extreme opinions, repetitive content, and offensive advertising. Through these feedback buttons, we can reduce the weight of these recommendations and even find and delete some bad community content.

Commercial companies should spend more energy to understand what their algorithms support and promote, rather than how to keep users on this platform. But it is clear and frustrating that this is not in their commercial interest. But ethically, this is imperative.

E. Deep model

In order to improve the accuracy of recommendation systems, deep learning techniques are increasingly used in the framework of recommendation systems. The reason why the recommendation system can generate a personalized recommendation list is based on the user's historical behavior information, but often the user's historical behavior is minimal and precious. Therefore, data sparseness has always been a challenge for recommendation systems. The current mainstream mitigation scheme is to combine side information in addition to the user-item interaction matrix, such as social information in the user dimension, document and image information in the item dimension, and context information (time, geographic information) in the user-item. [67] For the use of side information, artificial feature engineering can be used to extract the desired features, but because more information is combined, the cost of human labor is often too high. Fortunately, deep learning has its role in various types of side information, so that the system can hand over complex feature extraction to deep learning, and the recommendation system is more focused on better recommendation strategies. [18]

Due to the different sources of information for feature extraction, the methods used are completely different. Word vector method for text and convolutional neural network method for pictures. Due to space limitations, here only mentions a deep learning model that solves data sparse and cold start problems. The lack of valid data has always been the most serious problem facing recommendation systems. In order to solve this problem, the solution proposed by the industry is a cross-domain recommendation system, that is, the use of data from multiple domains to improve recommendation performance. EMCDR proposes an embedded mapping framework that can be used to optimize cross-domain recommendation problems. It mainly proposes two points: first, using hidden factor models for embedded learning, and second, using mapping techniques to solve cross-domain sparse problems. The Mapping technologies involved are mainly linear mapping and multilayer perceptron mapping (MLP). It is worth noting that because MLP can capture non-linear factors and excellent fitting ability, performance is due to linear mapping. [40] The specific model is shown in Figure 3. The application of deep

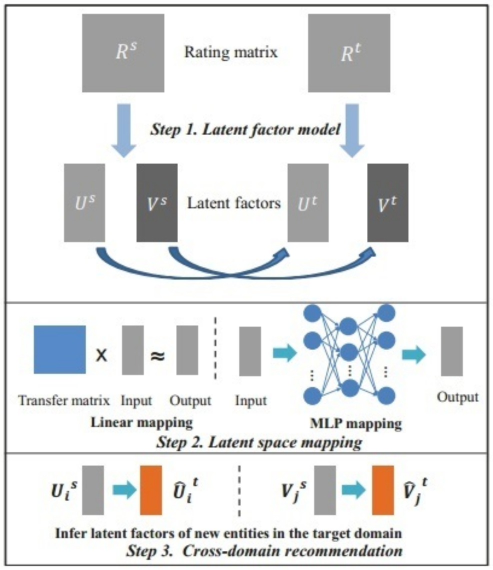

Fig. 3. EMCDR framework

learning technology in the recommendation system improves the data utilization efficiency of commercial companies and alleviates the cold start problem. The local modeling and feature extraction for individual users is gradually simplified due to the optimization of neural networks by hardware.

V. Conclusion

Aiming at the failure scenario of the recommendation system, this article focuses on solving two problems: how to improve the understanding of users? how to improve the quality of recommendations? Then the article summarized the following views:

- It is necessary to train offline models for individuals to adapt behavior habits and daily routines
- Increasing the angle of judging recommended content can effectively improve the quality of recommendations
- New technology can solve the problem of privacy leakage to a certain extent, but still needs legal norms and technical standards
- Conflicts between ethical issues and business goals of recommendation systems are growing
- Deep learning technology brings new ideas to the recommendation system, using less data to get better results

References

[1] Fakhri Abbas and Xi Niu. One size does not fit all: Modeling users' personal curiosity in recommender systems. *arXiv preprint arXiv:1907.00119*, 2019.

[2] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

[3] Charu C Aggarwal et al. *Recommender systems*, volume 1. Springer, 2016.

[4] Jan Philipp Albrecht. How the gdpr will change the world. *Eur. Data Prot. L. Rev.*, 2:287, 2016.

[5] Alessandro Bessi, Fabiana Zollo, Michela Del Vicario, Michelangelo Puliga, Antonio Scala, Guido Caldarelli, Brian Uzzi, and Walter Quattrociocchi. Users polarization on facebook and youtube. *PloS one*, 11(8), 2016.

[6] Robin Burke. Hybrid web recommender systems. In *The adaptive web*, pages 377–408. Springer, 2007.

[7] Pablo Castells, Neil J Hurley, and Saul Vargas. Novelty and diversity in recommender systems. In *Recommender systems handbook*, pages 881–918. Springer, 2015.

[8] Michael Castelluccio. The music genome project. *Strategic Finance*, pages 57–59, 2006.

[9] Tianfeng Chai and Roland R Draxler. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, 7(3):1247–1250, 2014.

[10] Kathleen Chaykowski. Facebook says data on 87 million people may have been shared in cambridge analytica leak. *Forbes. Retrieved April*, 4:2018, 2018.

[11] Li Chen, Yonghua Yang, Ningxia Wang, Keping Yang, and Quan Yuan. How serendipity improves user satisfaction with recommendations? a large-scale user evaluation. In *The World Wide Web Conference*, pages 240–250, 2019.

[12] Rui Chen, Qingyi Hua, Yan-Shuo Chang, Bo Wang, Lei Zhang, and Xiangjie Kong. A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks. *IEEE Access*, 6:64301–64320, 2018.

[13] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 39–46, 2010.

[14] Keith Decker, Katia Sycara, and Mike Williamson. Middle-agents for the internet. In *IJCAI (1)*, pages 578–583, 1997.

[15] Haley DeLeon. The ethical and privacy issues of recommendation engines on media platforms. https://towardsdatascience.com/the-ethical-and-privacy-issues-of-recommendation-engines-on-media-platforms-9bea7 Accessed Apr 24, 2019.

[16] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

[17] Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 371–380, 2009.

[18] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*, pages 278–288, 2015.

[19] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 257–260, 2010.

[20] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178, 2009.

[21] Damien Geradin and Dimitrios Katsifis. Taking a dive into google's chrome cookie ban. *Available at SSRN*, 2020.

[22] David K Gifford, Robert W Baldwin, Stephen T Berlin, and John M Lucassen. An architecture for large scale information systems. *ACM SIGOPS Operating Systems Review*, 19(5):161–170, 1985.

[23] David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[24] Cédric Gossart. Can digital technologies threaten democracy by creating information cocoons? In *Transforming politics and policy in the digital age*, pages 145–154. IGI Global, 2014.

[25] Khalid Haruna, Maizatul Akmar Ismail, Suhendroyono Suhendroyono, Damiasih Damiasih, Adi Cilik Pierewan, Haruna Chiroma, and Tutut Herawan. Context-aware recommender system: A review of recent developmental process and future research direction. *Applied Sciences*, 7(12):1211, 2017.

[26] Jon Haupt. Last. fm: People-powered online radio. *Music Reference Services Quarterly*, 12(1-2):23–24, 2009.

[27] David E Heckerman, John S Breese, Eric Horvitz, and David Maxwell Chickering. Collaborative filtering utilizing a belief network, December 30 1997. US Patent 5,704,017.

[28] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250, 2000.

[29] Thomas Hofmann. Probabilistic latent semantic analysis. *arXiv preprint arXiv:1301.6705*, 2013.

[30] Shuhui Jiang, Xueming Qian, Jialie Shen, Yun Fu, and Tao Mei. Author topic model-based collaborative filtering for personalized poi recommendations. *IEEE transactions on multimedia*, 17(6):907–918, 2015.

[31] Zach Jorgensen, Ting Yu, and Graham Cormode. Conservative or liberal? personalized differential privacy. In *2015 IEEE 31St international conference on data engineering*, pages 1023–1034. IEEE, 2015.

[32] Joseph A Konstan, Bradley N Miller, David Maltz, Jonathan L Herlocker, Lee R Gordon, and John Riedl. Grouplens: applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

[33] Yehuda Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 447–456, 2009.

[34] Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, pages 208–211, 2008.

[35] Dave Lee. Facebook founding president sounds alarm. https://www.bbc.co.uk/news/technology-41936791. Accessed 9 November 2017.

[36] Qing Li, Jia Wang, Yuanzhu Peter Chen, and Zhangxi Lin. User comments for news recommendation in forum-based social media. *Information Sciences*, 180(24):4929–4939, 2010.

[37] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.

[38] Ernst Lutz, Hans V Kleist-Retzow, and Karl Hoernig. Mafia—an active mail-filter-agent for an intelligent document processing support. *ACM SIGOIS Bulletin*, 11(4):16–32, 1990.

[39] Pattie Maes. Agents that reduce work and information overload. In *Readings in human–computer interaction*, pages 811–821. Elsevier, 1995.

[40] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. Cross-domain recommendation: An embedding and mapping approach. In *IJCAI*, pages 2464–2470, 2017.

[41] Frank McSherry and Ilya Mironov. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 627–636, 2009.

[42] Tamás Mihálydeák, Fan Min, Guoyin Wang, Mohua Banerjee, Ivo Düntsch, Zbigniew Suraj, and Davide Ciucci. *Rough Sets: International Joint Conference, IJCRS 2019, Debrecen, Hungary, June 17–21, 2019, Proceedings*, volume 11499. Springer, 2019.

[43] Silvia Milano, Mariarosaria Taddeo, and Luciano Floridi. Recommender systems and their ethical challenges. *AI & SOCIETY*, pages 1–11, 2020.

[44] Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G Azzolini, et al. Deep learning recommendation model for personalization and recommendation systems. *arXiv preprint arXiv:1906.00091*, 2019.

[45] Li-Hui Nian, Jing Wei, and Can-Bin Yin. The promotion role of mobile online education platform in students' self-learning. *International Journal of Continuing Engineering Education and Life Long Learning*, 29(1-2):56–71, 2019.

[46] Michiharu Niimi, Fuyuki Masutani, and Hideki Noda. Protection of privacy in jpeg files using reversible information hiding. In *2012 International Symposium on Intelligent Signal Processing and Communications Systems*, pages 441–446. IEEE, 2012.

[47] Kun Niu, Haizhen Jiao, Xiao Xu, Cheng Cheng, and Chao Wang. A novel learning approach to improve mobile application recommendation diversity. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1300–1307. IEEE, 2018.

[48] Carlos Ordonez, Il-Yeol Song, Gabriele Anderst-Kotsis, A Min Tjoa, and Ismail Khalil. *Big Data Analytics and Knowledge Discovery: 21st International Conference, DaWaK 2019, Linz, Austria, August 26-29, 2019, Proceedings*, volume 11708. Springer, 2019.

[49] Yoon-Joo Park and Alexander Tuzhilin. The long tail of recommender systems and how to leverage it. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 11–18, 2008.

[50] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[51] Stephen Pollock. A rule-based message filtering system. *ACM Transactions on Information Systems (TOIS)*, 6(3):232–254, 1988.

[52] Fulan Qian, Qianqian Min, Shu Zhao, Jie Chen, Xiangyang Wang, and Yanping Zhang. Three-way decision collaborative recommendation algorithm based on user reputation. In *International Joint Conference on Rough Sets*, pages 424–438. Springer, 2019.

[53] Zhan Qin, Ting Yu, Yin Yang, Issa Khalil, Xiaokui Xiao, and Kui Ren. Generating synthetic decentralized social graphs with local differential privacy. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 425–438, 2017.

[54] J. Ren, X. Xu, Z. Yao, and H. Yu. Recommender systems based on autoencoder and differential privacy. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, volume 1, pages 358–363, Jul 2019.

[55] Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[56] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Recommender Systems Handbook*, volume 1-35, pages 1–35. 10 2010.

[57] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.

[58] Francesco Ricci, Lior Rokach, and Bracha Shapira. Recommender systems: introduction and challenges. In *Recommender systems handbook*, pages 1–34. Springer, 2015.

[59] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.

[60] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.

[61] ACM SIGCHI. Recsys' 07: Proceedings of the 2007 acm conference on recommender systems. *Minneapolis, MN*, 2007.

[62] Brent Smith and Greg Linden. Two decades of recommender systems at amazon. com. *Ieee internet computing*, 21(3):12–18, 2017.

[63] Fagen Song, Tinghuai Ma, Yuan Tian, and Mznah Al-Rodhaan. A new method of privacy protection: Random k-anonymous. *IEEE Access*, 7:75434–75445, 2019.

[64] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. Major components of the gravity recommendation system. *Acm Sigkdd Explorations Newsletter*, 9(2):80–83, 2007.

[65] Omer Tene and Jules Polenetsky. To track or do not track: advancing transparency and individual control in online behavioral advertising. *Minn. JL Sci. & Tech.*, 13:281, 2012.

[66] Zeynep Tufekci. Youtube, the great radicalizer. https://www.nytimes.com/2018/03/10/opinion/sunday/youtube-politics-radical.html. Accessed March 10, 2018.

[67] Flavian Vasile, Elena Smirnova, and Alexis Conneau. Meta-prod2vec: Product embeddings using side-information for recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 225–232, 2016.

[68] David Vise. The google story. *Strategic Direction*, 2007.

[69] Cong Wang, Yifeng Zheng, Jinghua Jiang, and Kui Ren. Toward privacy-preserving personalized recommendation services. *Engineering*, 4(1):21–28, 2018.

[70] Hao Wang, Zonghu Wang, and Weishi Zhang. Quantitative analysis of matthew effect and sparsity problem of recommender systems. In *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pages 78–82. IEEE, 2018.

[71] Jianfang Wang and Pengfei Han. Adversarial training-based mean bayesian personalized ranking for recommender system. *IEEE Access*, 2019.

[72] Kexin Wang, Zhanfang Chen, YuanShuai Wang, and ZhangNan Yang. Feature fusion recommendation algorithm based on collaborative filtering. In *2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, pages 176–180. IEEE, 2019.

[73] Xing Wei, Xianglin Zuo, and Bo Yang. Sequential recommendation based on long-term and short-term user behavior with self-attention. In Christos Douligeris, Dimitris Karagiannis, and Dimitris Apostolou, editors, *Knowledge Science, Engineering and Management*, pages 72–83, Cham, 2019. Springer International Publishing.

[74] Yongxia Zhao and Bo Zhang. Research on recommendation algorithms based on collaborative filtering. In *Journal of Physics: Conference Series*, volume 1237, page 022094. IOP Publishing, 2019.

[75] Zhi-Dan Zhao and Ming-Sheng Shang. User-based collaborative-filtering recommendation algorithms on hadoop. In *2010 third international conference on knowledge discovery and data mining*, pages 478–481. IEEE, 2010.

[76] Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, 107(10):4511–4515, 2010.

[77] Andreas H Zins, Ulrike Bauernfeind, Fabio Del Missier, Adriano Venturini, and Hildegard Rumetshofer. *An experimental usability test for different destination recommender systems*. Citeseer, 2004.