Research Problems in Recommender systems

Nitin Mishra
Galgotias University, India
E-mail: drnitinmishral0@gmail.com

Saumya Chaturvedi
Galgotias University, India
E-mail: saumyanmishra5@gmail.com

Aanchal Vij
Galgotias University, India
E-mail: aanchal.vij04@gmail.com

Sunita Tripathi
University of Allahabad, India
E-mail: meet.suni@gmail.com

Abstract. With continuous growth of web applications around the globe, it is a challenge to find the suitable information needed for the user in a limited time. Number of handheld mobile devices is increasing and most of the business revolves around the correct search of the data. Without a proper recommender system it is very difficult to get required information from the web applications. Web applications use recommender systems to provide suitable data to users based on their choices and interests. For different kinds of needs different types of recommender systems have been proposed. Two most basic types of recommender systems are collaborative filtering recommender system and content based recommender system. Sometimes these two recommender systems are combined to increase the efficiency of a recommender system The generated new recommender system is known as hybrid recommender system.

The purpose of this paper is to help readers understand the basics of recommender systems. This paper identifies key areas of research openly available for new researchers. After reading this paper new researchers can understand basic problems of recommender systems which need improvement and hence they can make those problems their area of research.

1. Introduction
Recommender systems help us in getting the data which we need. It filters information which is are needed by the user. Today we have a lot of data in any system.[1, 2] The example of systems in which recommenders are needed is YouTube Netflix or any other E-Commerce platforms like Flipkart and Amazon. The scenario which we are facing today is that data is increasing and screen size is decreasing. Screen size we mean to say that initially systems were used from desktop and laptop which were around 15 inches and now it is being used by mobile
Figure 1. Model of recommendation process

which ranges from 4 to 7 inches. [3, 4, 5, 6, 7] When the user searches for any item and it is not available in first 5 or 10 searches he leaves the system and tries that item on another system. The biggest irony can be an item is available but didn’t come on the top of the searches which user did. so in this case what will happen is he will buy that item from another computer E-Commerce platform rather than the platform which was unable to provide a recommendation. A recommendation system can increase sales of a particular application. Many E-Commerce platforms did field because of lack of good recommendation system on their platforms. A good recommender system also staves users time and keep the user engaged in the system resulting in higher revenue. [8]

Many top-notch companies which are using recommendation systems are Google, YouTube Netflix, Flipkart, Amazon, Prime, gaana.com and many more. Every system comes with its advantages and disadvantages. So recommendation systems also face many problems which are yet to be solved effectively.

So the purpose of writing this paper is to make the Reader aware of recommender systems and its major techniques. The paper will also explore research problems in recommender systems based on the extensive study done by us using the papers which are referenced in the reference section. In this paper, we have identified some key areas of research which are open to new researchers. so students who are in their Masters and PhD can take this area and take the topic as their area of research and contribute to the development and improvement of recommender systems of the new generation.

We have included papers from early 1997 to 2020. More than 50 papers have been included in the study. We have not only identified problems but the latest solutions to problems papers have also been added for researchers to understand the problem in detail.

2. GENERAL CONCEPTS

Recommender systems is a system that helps users to choose items which they may need. different artificial intelligence techniques and machine learning techniques are applied to achieve this output. [9, 10]

Some examples of recommender systems can be google.com, amazon.com, Netflix, and other popular e-commerce, music and video portals available online. Since these systems have millions of items so they can not function properly without a good recommender system.

In a recommendation-system application, there are two classes of entities, which we shall
refer to as users and items. The formal definition of the recommender system is:

- C: The set of all users
- S: The set of all possible items that can be recommended, for example, video, songs and books.
- U: A utility function that measures the usefulness of a specific item s ∈ S to user c ∈ C, i.e., U: C × S → R, where R is a totally ordered set.

The space S of feasible items can be very large, ranging in lots of hundreds or even thousands and thousands of items in some applications, such as recommending books or CDs. Similarly, a person’s area can also be very large—millions in some cases.[11]

In recommender machine how useful the item is determined by its rating. Rating is a measure of item liked by the user. As it is given by the user hence reliability of this rating has extra value in understanding the choices of the user. The rating can be taken in different ways. some common forms of rating are it is asked based on 1 to 5 scales as in case of apps in Google Play. It can also ask based on the scale of 1 to 10 as used by many rating methods. Many customer service agencies use this scale of rating. Whatever rating scale you are going to use your one and describes like and others end describes the extent of dislike.[12]

User profile can be generated by storing its traits like age gender area email mobile and other things. Item profile can be generated based on features of the item. Like in case of a book it is the language author Jonah cost publisher etc. In the case of television, it is the brand feature power consumption and many features which can be used to create the profile. The way we create the profile has an important impact on the recommendation system.[13] Ratings are done on a subset of data rather than entire data. Rating matrix is created between user and item and this becomes the heart of the recommender system. The way the rating matrix is analysed defines the recommender system. Different domains use Different techniques for extracting data from the user-item matrix. [14, 15, 5]

Recommender systems are recommends different items to the user based on whatever machine learning or artificial intelligence techniques they have used on the rating matrix.[15] Good recommender systems improve with user feedback. A good recommender system also performs good recommendation even though very few ratings are there. The accuracy of a good recommender systems increases as the history of user increases in the system. [16]

3. TECHNIQUES USED IN RECOMMENDATION SYSTEM

The Recommender systems are categorised on the basis of rating techniques used by the system. The way a systems takes ratings or predicts user preferences defines the Recommender systems. Figure 2 describes the basic techniques used in recommender systems. In the coming sections, we are going to describe them. In this paper, we have given a brief description of the basic techniques used in recommender systems. Other techniques are also there which are combined to generate hybrid recommender systems. Figure 4 describes them.

3.1. Collaborative Filtering

The collaborative filtering system is a system where we take input from the user and use those inputs to create relations with other users and items.[10, 14, 15]

For example, any user in a collaborative filtering system is asked to rate a particular item. Similarly, other users also rate different items. In this way, we get a user-item rating matrix. The rating done by the user gives historical data and choices of the user. Through choices of users, we can create a profile of the user. Then we can calculate what new items we can purchase on the basis of his history and his profile.
Some examples of collaborative filtering systems are movie recommendation systems and social networks.[4]

3.2. Content based
Content-based system requires the item to the user based on the description of the item rather than the history of the user.[9, 17] These types of recommender systems are used in newspapers and article recommendation systems. These recommender systems may have little information about the user. Figure 3 explains the process of recommendation.

These recommender systems do not have a history of the user. The most common technique used by content-based recommender systems is item profiling. This popularity of the item is recorded. User profiling is also done sometimes with little basic information about the user which system has. Item profiles are based on the properties of items. The recommender system uses the properties to recommend items to different users. Unlike collaborative filtering recommender systems, they do not have rating information. User profiles can also be created from the likes and dislikes of users to a particular item. [9, 17]

When we want to recommend any item to the user, user interests are compared against item properties. This is also known as content features of items which includes the following:[6, 18]

- The system contains a big database of the item to be recommended. This database consists of features of items. This database is known as an item profile database.
- Users provide little information about their preference likes and dislikes to the system and with this little information the system builds a user profile.
- The recommendation is done on the basis of a comparison of item profile with user interest.

One can make better-personalised recommendations by means of utilising the elements of gadgets and users. An object profile is defined by way of its essential features. For example, a book can be described using its title, genre, language, publisher, cost etc. Using the weighting procedure, similarity can be calculated between items. In some domains, we can represent elements by means of boolean values while in others we can represent the values using a set of restrained values. Consider the example of the newspaper where we analyze the newspaper articles on the basis of the exceptional form of topics. Boolean cost is indicative of whether a phrase is present...
Figure 3. Content based Process

in the article or not. Integer cost may want to define the categorical way the range of time of word appears in an Article. This method gives a successful recommendation in content-based recommender systems without using explicit ratings.[19, 20]

3.3. Hybrid Recommendation System
This system recommends on the basis of the mixture of some techniques. This is one of the recent trends in the recommendation system. The table given shows different types of recommendation techniques. The Hybrid makes use of a combination of these recommendation techniques. The recommendation system by Facebook seems to make use of all these techniques. The hybrid recommendation takes advantage of all the techniques but once should be very careful as hybrid may involve a lot of computation and may give conflicting results.[6, 21, 22]

Figure 4 describes some popular techniques combined to generate hybrid recommender systems. These techniques are combined with each other when they improve the results of the recommendation.

4. RESEARCH PROBLEMS
These recommendation systems have a great future. Today some problems are yet to be solved by the research community to make research more efficient. Some of the problems which we understand can be solved are listed below. Figure 5 Shows all the research problems we are going to address in the later sections. These research problems can give ideas to work in the area.

4.1. Gathering Known Ratings for Matrix
It has been observed that most of the users do not give any ratings. So a research problem arises that how to know whether they are satisfied with the product and how much. There are two ways of taking ratings. When is explicit like asking them after they have purchased or gone through any item. Another way which is predicting their ratings for a particular item based on their preferences on some other item. This method is known as an implicit method of collecting ratings. According to our research, explicit and implicit methods have sufficient gap which gives an opportunity for researchers to start research in this domain.[24, 25]

4.2. Cold Start Problem
A cold start problem is a problem that arises when no information is found about the user or item in the system.[7] collaborative filtering recommender system which needs mandatory information
| Recommendation Technique | Environment | Input to System | Method |
|--------------------------|-------------|----------------|--------|
| Collaborative            | Rating from users of items | Rating Unit | Identify user’s rating and recommend accordingly. |
| Content - Based          | Features of items | User’s rating of items | Classify user’s rating / behavior |
| Demographic              | Demographic information of user and their rating | Demographic information about the user | Recognize users which are demographically similar to other users |
| Utility – Based          | Properties/Features of items | A utility function over item that describe the user properties | Apply the function over the item and determine the rank |
| Knowledge – Based        | Knowledge of how these items meet the user’s requirement | A account of user’s need of interest | Deduce the match between item and users need |

**Figure 4.** Techniques used by Hybrid Recommendation[23, 4, 21]

**Figure 5.** Research Problems in Recommender systems
about user and item before recommending fails. We can have three different subproblems in the domain of cold start problems. The first problem arises when we do not have any information about the new user who is entering the system. this only happens when a user joins the system for the first time. An example can be if you are joining Amazon or Flipkart for the first time this kind of problem arises. This problem is known as the new user cold-start problem.[26] The second problem arises when we introduce a new item to the system. this item is very original of its kind. The recommender system is unable to find any ratings associated with this item. The collaborative filtering system which needs user-item rating Matrix in order to give a recommendation is unable to start and the problem is known as cold start item problem. The third problem arose when we launched the system for the first time. In this case, we do not have either any user information or any item information. In other words, we do not have any user-item matrix of ratings which is required for collaborating recommender systems to work properly. This problem is known as the cold start system problem. In cold start problems, famous content-based solutions can be applied to solve the problem.[27, 28] Other solutions are using a combination of various machine learning techniques.[29, 30]

4.3. Sparsity Problem

In the recommendation system, it has been noticed that most users use the system but do not give rating for feedback to the system in a proper way. So even though we may have many users using the recommender system various Times, it is possible that we have very few ratings from those users about different items which they have liked or purchased or even disliked. Rating seems useless to users, so they avoid giving it and sometimes may lead to false rating such as providing 5 stars( Considered Best for them) or 1 stars (Considered non likely item), without even noticing what kind of product it is. This is taken as input by the recommender system, which further displays the unwanted results to the user, which may lose the interest of the user in that platform, and lead to non-efficient working. As rating has no significance as per the users, they sometimes don’t even give to bulk of products, which again leads to the same problem. [31]

Sparsity problem is a problem which arises due to sparsity of rating matrix. here we are talking about the user-item rating matrix. In mathematical terms user-item rating matrix is sparse. It gives rise to a unique problem known as sparsity problem in recommender systems.[32, 33].

This problem is the problem of collaborative filtering recommender systems domain. This problem gives unique opportunities to new researchers to find new ways to predict this missing data. [34]

4.4. Scalability

Scalability is the property of the system which defines weather system will be able to cope up when the system grows. [34] For example, in case of recommender systems, scalability can be understood as a situation where a recommender system is performing very well in case of few users like 1000 users but as the user grows to 10000 or 100000 it starts performing a way which is not desirable. When the system faces scalability issues it becomes slow it starts feeling it start giving problems which it has never given when a load of users recommendation were less.

The scalability issues can be divided into two parts hardware scalability and software scalability. the hardware is scalability is about the increase of hardware to solve the scalability problem. For example, one can increase processor, RAM and server configuration to solve the problem. But only hardware increase cannot solve the problem.[34]

Software scalability is about writing algorithms and using methods which work well when hardware configuration is increased as needed in future. although this is a major problem which is not as easy as it seems to be. because there are algorithms which perform very well when the amount of data on which they have to operate is small but as the data increases they start
performing inefficiently. The accuracy of the prediction decreases as the data increases. Some algorithms are not able to utilise the increased efficiency of the hardware hence create a problem of scalability.[32]

So this is also an open area of research in recommender systems. As we know incoming Technology we are going for parallel processing and Hadoop and other architectures and big data. So this problem needs to be addressed. So it gives upcoming researchers a new area to start the research and solve this problem using an innovative method.[33]

4.5. Over Specialization problem
This problem arises if the recommended items are too similar to each other. One such example is that a user buys grocery items from a shopping website on the regular basis, and whenever he/she opens the website, the recommender system recommends only sugar, may be of different brands (basis on the maximum purchase), on seeing same item again and again and not any unique suggestion may lead user to switch the shopping website whose recommender system offers better, interesting and personalized results for the user. So in this problem basically, the user loses interest in recommended items, one solution to this problem is recommendation diversification. In this method we list all the items that are dissimilar but it is of interest to the user[3].

4.6. Lack of Data
Perhaps the biggest issue facing recommender systems is that they need a lot of data to effectively make recommendations. In the world of the recommender system, it is common practice to use a publicly available dataset from a different environment in order to evaluate the efficiency of recommendation algorithms. [35] These data sets are very important and are used as a benchmark to develop new recommendation algorithms. Most of the top companies like Google, Amazon, make good recommendation because they are having a lot of consumer user data so recommender system firstly needs consumer/item user data (from different sources), then it must perform some statistical analysis based on some procedure (User behaviour observation or events), and then the Recommendation algorithm does its work. If we will be having more consumer/item data we will be in a situation with the help of the recommender algorithm to have better recommendations[36, 37].

4.7. Changing data
The most important challenge in the case of recommender systems is changing data i.e. that they are usually subjective towards the old/past and face problems while dealing with new or recommending new. Many recommendation systems collect user data or make use of user past behavior but depending on user past behaviour analysis is not an efficient tool for a good recommendation because users trends and interests are always changing so it would become very difficult for the recommendation system to react in a changing environment of data. To show you what I mean, usually a newlywed couple checks for dresses, beauty products and lifestyle items but with the passage of time and change of situations their interest changes to baby products such as diapers, baby foods etc, which is entirely different from the previous search which may confuse the recommender system and may lead to display results which are no more interest of the buyer. [38]

4.8. Changing user preferences
The next important challenge in the recommender system is changing user preferences and making coordination with rapidly changing preferences of the user. As we observed till now that user intention towards browsing some item may be different at different interval of time so
recommendation systems which are completely based on user preferences may perform wrong recommendations for eg. Suppose today I am browsing books for myself but tomorrow I may browse sports item for myself or It can be said that a 10 year kid is searching for multiple items without any thought just to scroll without any intention of purchase, at that time the recommendation system based on user preferences may recommend me the wrong items, so making coordination with frequently changing user preferences is the most important issue in recommendation system[38].

4.9. Unpredictable items
The next important challenge in the recommender system is unpredictable items based on different rankings. Item recommender used to find out the ranked list of predictable list of items and unpredictable list of items. It may include unusual products which existed before or have relevance with the existing ones, in this scenario the system may not be able to recommend the product to any of the users, and thus will be dumped for ages, considering Amazon launched a new product for selling, i.e AI based Robot. Since this belongs to all new category, so no browsing history of the category found which leads to no recommendations for the user. In the same manner, movie recommendations fall into this category of the recommendation of unpredictable items. Content-based filtering technique addresses this problem.[39]

5. Conclusion
This paper has introduced recommender systems to new researches. This paper has also identified key problems which need research in recommender systems. This paper can help PhD and Masters students in choosing their area of research. The research gap is already presented in this paper to form different problems of recommender systems.

The recommendation system finds its utility in major areas of web Applications. As these problems get solved more and more useful recommendation systems will become. With more reliable recommendations web applications will be more intelligent and usable.

References
[1] G. Adomavicius and A. 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, vol. 17, no. 6, pp. 734–749, 2005.
[2] D. Agarwal and B.-C. Chen, “Regression-based latent factor models,” in Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’09, (New York, NY, USA), pp. 19–28, ACM, 2009.
[3] P. Adamopoulos and A. Tuzhilin, “On over-specialization and concentration bias of recommendations: Probabilistic neighborhood selection in collaborative filtering systems,” in Proceedings of the 8th ACM Conference on Recommender systems, pp. 153–160, ACM, 2014.
[4] M. Jiang, P. Cui, F. Wang, Q. Yang, W. Zhu, and S. Yang, “Social recommendation across multiple relational domains,” in Proceedings of the 21st ACM international conference on Information and knowledge management, pp. 1422–1431, ACM, 2012.
[5] X. Su and T. M. Khoshgoftaar, “A survey of collaborative filtering techniques,” Adv. in Artif. Intell., vol. 2009, pp. 4:2–4:2, Jan. 2009.
[6] A. Gunawardana and C. Meek, “A unified approach to building hybrid recommender systems,” in Proceedings of the Third ACM Conference on Recommender Systems, RecSys ’09, (New York, NY, USA), pp. 117–124, ACM, 2009.
[7] Y. Rong, X. Wen, and H. Cheng, “A monte carlo algorithm for cold start recommendation,” in Proceedings of the 23rd international conference on World wide web, pp. 327–336, ACM, 2014.
[8] Y. Zhang, F. Sun, X. Yang, C. Xu, W. Ou, and Y. Zhang, “Graph-based regularization on embedding layers for recommendation,” ACM Trans. Inf. Syst., vol. 39, Sept. 2020.
[9] Z. Huang and M. K. Ng, “A fuzzy k-modes algorithm for clustering categorical data,” IEEE Transactions on Fuzzy Systems, vol. 7, no. 4, pp. 446–452, 1999.
[10] M. Papagelis, D. Plexousakis, and T. Kutsuras, “Alleviating the sparsity problem of collaborative filtering using trust inferences,” in Trust management, pp. 224–239, Springer, 2005.
[11] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme, “Learning attribute-to-feature mappings for cold-start recommendations,” in Proceedings of the 2010 IEEE International Conference on Data Mining, ICDM ’10, (Washington, DC, USA), pp. 176–185, IEEE Computer Society, 2010.
[12] N. Houlsby, J. M. Hernández-Lobato, and Z. Ghahramani, “Cold-start active learning with robust ordinal matrix factorization,” in International Conference on Machine Learning, pp. 766–774, 2014.
[13] N. N. Liu, X. Meng, C. Liu, and Q. Yang, “Wisdom of the better few: Cold start recommendation via representative based rating elicitation,” in Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys ’11, (New York, NY, USA), pp. 37–44, ACM, 2011.
[14] N. Golbandi, Y. Koren, and R. Lempel, “Adaptive bootstrapping of recommender systems using decision trees,” in Proceedings of the fourth ACM international conference on Web search and data mining, pp. 595–604, ACM, 2011.
[15] Y. Shi, M. Larson, and A. Hanjalic, “Tags as bridges between domains: Improving recommendation with tag-induced cross-domain collaborative filtering,” in International Conference on User Modeling, Adaptation, and Personalization, pp. 305–316, Springer, 2011.
[16] F. Ricci, L. Rokach, and B. Shapira, “Introduction to recommender systems handbook,” in Recommender systems handbook, pp. 1–35, Springer, 2011.
[17] M. J. Pazzani and D. Billsus, “Content-based recommendation systems,” in The adaptive web, pp. 325–341, Springer, 2007.
[18] M. Zhang, J. Tang, X. Zhang, and X. Xue, “Addressing cold start in recommender systems: A semi-supervised co-training algorithm,” in Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR ’14, (New York, NY, USA), pp. 73–82, ACM, 2014.
[19] B. Li, “Cross-domain collaborative filtering: A brief survey,” in Tools with Artificial Intelligence (ICTAI), 2011 33rd IEEE International Conference on, pp. 1085–1086, IEEE, 2011.
[20] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, “Methods and metrics for cold-start recommendations,” in Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’02, (New York, NY, USA), pp. 253–260, ACM, 2002.
[21] S.-T. Park, D. Pennock, O. Madani, N. Good, and D. DeCoste, “Naïve filterbots for robust cold-start recommendations,” in Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’06, (New York, NY, USA), pp. 699–705, ACM, 2006.
[22] S.-T. Park and W. Chu, “Pairwise preference regression for cold-start recommendation,” in Proceedings of the Third ACM Conference on Recommender Systems, RecSys ’09, (New York, NY, USA), pp. 21–28, ACM, 2009.
[23] S. Sahebi and W. W. Cohen, “Community-based recommendations: a solution to the cold start problem,” Proceedings of WOODSTOCK ’97, 1997.
[24] S. Jaffali, S. Jamoussi, K. Smaill, and A. B. Hamadou, “Like-tasted user groups to predict ratings in recommender systems,” Social Netw. Analys. Mining, vol. 10, no. 1, p. 42, 2020.
[25] K. Zhou, S.-H. Yang, and H. Zha, “Functional matrix factorizations for cold-start recommendation,” in Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’11, (New York, NY, USA), pp. 315–324, ACM, 2011.
[26] X. Zhang, J. Cheng, T. Yuan, B. Niu, and H. Lu, “Semi-supervised discriminative preference elicitation for cold-start recommendation,” in Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management, CIKM ’13, (New York, NY, USA), pp. 1813–1816, ACM, 2013.
[27] P. Mazumdar, B. K. Patra, and K. S. Babu, “Cold-start point-of-interest recommendation through crowdsourcing,” ACM Trans. Web, vol. 14, Aug. 2020.
[28] N. K. Mishra, V. Mishra, and S. Chaturvedi, “Solving cold start problem using mba,” in 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), pp. 1598–1601, 2017.
[29] N. Mishra, V. Mishra, and S. Chaturvedi, “Tools and techniques for solving cold start recommendation,” in Proceedings of the 1st International Conference on Internet of Things and Machine Learning, IML ’17, (New York, NY, USA), Association for Computing Machinery, 2017.
[30] M. Saveski and A. Mantrach, “Item cold-start recommendations: Learning local collective embeddings,” in Proceedings of the 8th ACM Conference on Recommender Systems, RecSys ’14, (New York, NY, USA), pp. 89–96, ACM, 2014.
[31] N. Mishra, S. Chaturvedi, V. Mishra, R. Srivastava, and P. Bargah, “Solving sparsity problem in rating-based movie recommendation system,” in Computational Intelligence in Data Mining (H. S. Behera and D. P. Mohapatra, eds.), (Singapore), pp. 111–117, Springer Singapore, 2017.
[32] A. da Costa, E. Fressato, F. Neto, M. Manzato, and R. Campello, “Case recommender: A flexible and
extensible python framework for recommender systems," in Proceedings of the 12th ACM Conference on Recommender Systems, RecSys ’18, (New York, NY, USA), p. 494–495, Association for Computing Machinery, 2018.

[33] T. Kitazawa and M. Yui, “Query-based simple and scalable recommender systems with apache hivemall,” in Proceedings of the 12th ACM Conference on Recommender Systems, RecSys ’18, (New York, NY, USA), p. 502–503, Association for Computing Machinery, 2018.

[34] W. Pan, E. W. Xiang, N. N. Liu, and Q. Yang, “Transfer learning in collaborative filtering for sparsity reduction.,” in AAAI, vol. 10, pp. 230–235, 2010.

[35] Z.-K. Zhang, C. Liu, Y.-C. Zhang, and T. Zhou, “Solving the cold-start problem in recommender systems with social tags,” EPL (Europhysics Letters), vol. 92, no. 2, p. 28002, 2010.

[36] B. Lika, K. Kolonvatso, and S. Hadjiehtymiades, “Facing the cold start problem in recommender systems,” Expert Systems with Applications, vol. 41, no. 4, pp. 2065–2073, 2014.

[37] M. Yan, J. Sang, T. Mei, and C. Xu, “Friend transfer: Cold-start friend recommendation with cross-platform transfer learning of social knowledge,” in Multimedia and Expo (ICME), 2013 IEEE International Conference on, pp. 1–6, IEEE, 2013.

[38] N. Lathia, S. Hailes, L. Capra, and X. Amatriain, “Temporal diversity in recommender systems,” in Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, pp. 210–217, ACM, 2010.

[39] E. Herder and B. Zhang, “Unexpected and unpredictable: Factors that make personalized advertisements creepy,” in Proceedings of the 23rd International Workshop on Personalization and Recommendation on the Web and Beyond, ABIS ’19, (New York, NY, USA), p. 1–6, Association for Computing Machinery, 2019.

[40] M. Sun, F. Li, J. Lee, K. Zhou, G. Lebanon, and H. Zha, “Learning multiple-question decision trees for cold-start recommendation,” in Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, WSDM ’13, (New York, NY, USA), pp. 445–454, ACM, 2013.

[41] L. Zhang, D. Agarwal, and B.-C. Chen, “Generalizing matrix factorization through flexible regression priors,” in Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys ’11, (New York, NY, USA), pp. 13–20, ACM, 2011.

[42] D. Agarwal and B.-C. Chen, “Ilda: Matrix factorization through latent dirichlet allocation,” in Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM ’10, (New York, NY, USA), pp. 91–100, ACM, 2010.

[43] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme, “Learning attribute-to-feature mappings for cold-start recommendations,” in Data Mining (ICDM), 2010 IEEE 10th International Conference on, pp. 176–185, IEEE, 2010.

[44] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “Bpr: Bayesian personalized ranking from implicit feedback,” in Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence, pp. 452–461, AUAI Press, 2009.

[45] O. Moreno, B. Shapira, L. Rotkach, and G. Shani, “Talmud: Transfer learning for multiple domains,” in Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM ’12, (New York, NY, USA), pp. 425–434, ACM, 2012.

[46] B. Li, Q. Yang, and X. Xue, “Transfer learning for collaborative filtering via a rating-matrix generative model,” in Proceedings of the 26th Annual International Conference on Machine Learning, ICML ’09, (New York, NY, USA), pp. 617–624, ACM, 2009.

[47] H.-N. Kim, A. El-Saddik, and G.-S. Jo, “Collaborative error-reflected models for cold-start recommender systems,” Decision Support Systems, vol. 51, no. 3, pp. 519–531, 2011.

[48] H. J. Ahn, “A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem,” Information Sciences, vol. 178, no. 1, pp. 37–51, 2008.

[49] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, “A collaborative filtering approach to mitigate the new user cold start problem,” Knowledge-Based Systems, vol. 26, pp. 225–238, 2012.

[50] M.-H. Nadimi-Shahraki and M. Bahadorpouro, “Cold-start problem in collaborative recommender systems: Efficient methods based on ask-to-rate technique,” Journal of computing and information technology, vol. 22, no. 2, pp. 105–113, 2014.