Recommendation Systems in Real Applications: Algorithm and Parallel Architecture

Mengxian Li, Wenjun Jiang(✉), and Kenli Li

School of Information Science and Engineering, Hunan University, Changsha 410082, China
Jiangwenjun@hnu.edu.cn

Abstract. Recommendation systems are popular both in business and in academia. A series of works have been reported. In this paper, we briefly introduce the background and some basic concepts of recommendation systems, especially the applications in mainstream websites, most of them built upon parallel processing systems. However, how the recommendation algorithm works in real applications? We investigate (1) the key ideas of recommendation algorithms that are being used in real applications and (2) the parallel architecture in those real recommendation systems. In addition, the performance of recommendation system for those sites are also being analyzed and compared. We also analyze their features and compare their performances. Finally, we outline the challenges and opportunities that all recommendation systems are facing. It is anticipated that the present review will deepen people's understanding of the field and hence contribute to guide the future research of recommendation systems. Our work can help people to better understand the literature and guide the future directions.

Keywords: Recommendation system · Real application · Parallel architecture · Google news · Netflix · Meituan · Facebook

1 Introduction

With the development of the internet and information technology, we have entered the era of great explosion of the information. This phenomenon leads to big challenges for both resource consumers and providers. Recommendation system is mainly composed of three parts: the input (e.g., user preference), recommendation process (i.e., finding out the information or commodity user may be interested in) and the output (i.e., showing the recommendation result). In this article, we focus on investigating recommendation algorithms which are actually running in business applications, mainly from two aspects: the key idea of the algorithm and the parallel architecture of the real system.

At the year of 1992, Goldberg et al. [1] proposed the idea of collaborative filtering in the Tapestry system of Palo Alto Research Center for the first time. In 1994, Resnick et al. [2] first proposed the use of collaborative filtering algorithm

G. Wang et al. (Eds.): SpaCCS 2016, LNCS 10066, pp. 45–58, 2016.
DOI: 10.1007/978-3-319-49148-6_5

to filter Network News. Therefore, GroupLens become one of the first automated collaborative filtering recommendation system. In the end of 20th century, the e-commerce site which represented by Amazon appeared and promoted the development of the recommendation system. As a technology, recommendation system has been widely used in various disciplines. Therefore, techniques in recommendation systems are developing rapidly. They can be classified into several categories, including content-based recommendation [3,4], knowledge-based recommendation [5,6], collaborative filtering recommendation (CF) [7,8], etc. Meanwhile, in the wake of new technologies such as parallel computing [9], data mining [10] and so on, some other new methods are being developed, including trust-aware recommendation [11], location-based recommendation [12], time-dependant recommendation [13], and so on. Since each method has its own pros and cons, a more common way is to combine several methods to produce more effective results [14].

Recommendation system has been studied for more than twenty years with a lot of relevant reports. However, it lacks a comprehensive study on recommendation systems from the perspective of real application. This motivates our work in this paper. We strive to study recommendation systems that are actually running in business applications. Our contributions are threefold, as follows:

1. We selectively study several representative real applications in which recommendation algorithms run as key components. We investigate (1) the key ideas of recommendation algorithms that are being used in those applications and (2) the parallel architecture in those real recommendation systems.
2. We comprehensively compare the representative recommendation systems from multiple aspects. It helps us to better understand the literature and guide the future directions for both researchers and application designers.
3. Based on the above two works, we make a further step to point out the current research hotspots, the remaining open challenges, and the promising research directions of recommendation systems.

2 Related Work

As a project, which is popular in both commercially and in terms of the academic research, many scholars have proposed various researches on recommendation. On taxonomy of recommendation systems, in [15], Schafer et al. presented an explanation of how recommendation system help online retailers increase income. Based on six real world examples, they created a taxonomy of recommendation systems from several aspects. Two years later, they further created a new taxonomy of recommendation systems and published another survey [16] to introduce the additional knowledge required from the database, ways of recommendations presented, and different level of personalization. In addition, they identified five commonly used E-commerce recommendation application models. In 2014, Bao et al. [17] proposed three taxonomies according to data source, method, and objective. They also summarized the goals, contributions and comparative analysis for each category. Moreover, in 2016, Jiang et al. [18] present

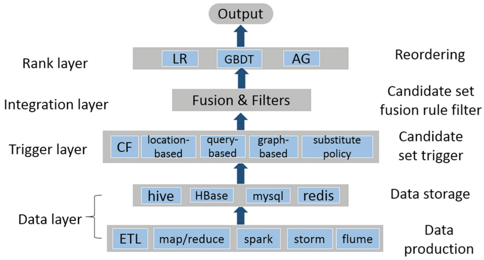

Fig. 1. The framework of Meituan

a comprehensive survey on graph-based trust evaluation models, in which trust-based recommendation is taken as an important application.

The implementation methods of parallel recommendation can be categorize as the distributed implementation, the parallel implementation, the platform-based implementation and the heterogeneous implementation of memory-based, model-based and hybrid recommendation systems [19]. Liang et al. [20] proposed a parallel user profiling approach, employing the advanced cloud computing techniques, Hadoop, MapReduce and Cascading. Christou et al. introduced a parallel multi-threaded implementation of collaborative filtering combined with a custom content-based algorithm in [21], which outperformed state-of-the-art implementations of similar algorithms.

3 Representative Recommendation Systems in Real Applications

Nowadays, many real applications are taking advantage of recommendation algorithms. In this section, we focus on key ideas and parallel architectures of the recommendation systems in real applications.

3.1 E-commerce Application: Meituan.com (2015)

Meituan is the first Groupon e-commerce site in China. It helps consumers find the most trustworthy businesses and low discount products. In 2015, the technology team of Meituan.com published an article in their official website[1], describing some of the practices about how they build and enhance their recommendation system.

Key idea: Figure 1 illustrates the framework of Meituan's recommendation system. There are mainly four layers: the data layer, the trigger layer, the integration layer, and the rank layer.

Details: As the name suggests, the data layer generates and stores the data. The trigger layer explores several triggering policies to produce recommendation candidate set, according to historical behavior, real-time behavior, location

[1] http://tech.meituan.com/mt-recommend-practice.html.

and other information of users. In the trigger layer, a variety of algorithms are explored simultaneously to improve the recommendation quality. After this, in the integration layer, it uses modulation and classification to fuse the results of each recommendation method and filter out unwanted items. In the last layer, it uses machine learning model to reorder the candidate sets, which are selected by the trigger layer before. This solves the ranking issues when fusing candidate set of different strategies.

Details of similarity calculating: Particularly, the similarity in the collaborative filtering, is calculated by loglikelihood ratio, which is used as an similarity calculation method in mahout. Assume that there are two events, A and B, making statistics with their occurring frequency. Make $K_{1,1}$ represent the frequency of A and B occurred at same time, $K_{1,2}$ represent the frequency of B occurs but A doesn't, and so on for $K_{2,1}$ and $K_{2,2}$. Then, the loglikelihood ratio can be calculated by Formula 1.

$$Ratio = 2 * (matrixEn - rowEn - columnEn) \tag{1}$$

Here, $matrixEn$, $rowEn$ and $columnEn$ are calculated by employing Formula 2. Entropy here means the shannon entropy of the system composed of several elements.

$$\begin{cases} matrixEn = entropy(k_{1,1}, k_{1,2}, k_{2,1}, k_{2,2}) \\ rowEn = entropy(k_{1,1}, k_{1,2}) + entropy(k_{2,1}, k_{2,2}) \\ columnEn = entropy(k_{1,1}, k_{2,1}) + entropy(k_{1,2}, k_{2,2}) \end{cases} \tag{2}$$

Parallel approach: In order to provide real-time computing for a large amount of users, Meituan.com adopts technologies of parallel computing, load balancing, and real-time streaming data processing (by a speech of Hao Cao, the senior technical experts of Meituan.com). For example, they designed a FeatureLoader module[2], which accesses and computes features in parallel. In real applications, the average response time in parallel is about 20 ms faster than that in serial.

3.2 Social Network Application: Facebook (2015)

The friend recommendation function, as a very popular and practical personalized service in social network, aims to recommend new friends to users according to their history. Facebook is an online social network site which has its headquarter in USA. In addition to text messages, users can send pictures, videos, sound media messages and other types of files to their friends by loading Facebook. Facebook announced the principles, performance and usage of its recommendation system[3] in their official website.

[2] http://tech.meituan.com/meituan-search-rank.html.

[3] https://code.facebook.com/posts/861999383875667/recommending-items-to-more-than-a-billion-people/.

Key idea: Facebook uses a distributed iterative image processing platform–Apache Giraph[4]. It is able to support large-scale data, and thus is taken as the basis of its recommendation system. The training models take the combination of data parallel and model parallel.

Key problem: As introduced in their official website, recommendation system of Facebook uses CF and Matrix Factorization (MF). Stochastic gradient descent (SGD) [22,23], alternating least squares (ALS) [24] and other iterative algorithms are implemented to reduce the time and space complexity. To take advantages of these algorithms, standard method of Giraph[5] need to be improved. The standard method processes users and items as vertices and the rating as the edge weight between two vertices. The iteration process of SGD or ALS is to traverse all edges, and send feature vectors of vertices to each vertex, then update these feature vectors partially. However, there are several serious issues in this method. First, the iterative process will bring a huge network traffic load. Second, the different popularity of items could result in an uneven distribution of the node degrees, which may lead to insufficient memory or processing bottlenecks.

Solution: To address the above issues, Facebook invents an efficient and convenient method which explores work-to-work information transmission. It divides the original graph into N workers, which are linked end-to-end to form a worker circle. Each worker contains a collection of items and a number of users. At each step, the adjacent worker sends message (e.g., the updating information of items) clockwise to the next worker. Only the internal ratings of each worker are processed in each step, so that all ratings are processed after N steps. The traffic is independent with the number of ratings. Moreover, the second problem above does not exist any longer, because items are not represented by vertices any more. To further improve the performance, Facebook incorporates the two algorithms, ALS and SGD, and adopts a rotation hybrid solution.

Parallel approach: Facebook uses multi-GPU training parallel framework, and uses the combination of data parallel and model parallel training models. The multi-core architecture of GPU (Graphic Process Units) consists of thousands of stream processors, which operate in parallel and reduce the computation time dramatically. Data parallel means that it cut the training data into N parts, and train them by N-workers in parallel. Meanwhile, model parallel splits the model into several model units, which work together to complete the training.

Friend recommendation differs from other types of recommendation, in that users may not need the most popular users, but the ones who are more possible to be their friends. In other words, the cold-start effect is more serious than that in other types of recommendation. We suggest that the location-based recommendation and cross-platform recommendation can be incorporated to alleviate this problem.

[4] http://giraph.apache.org/.
[5] https://zh.wikipedia.org/zh-cn/Giraph.

3.3 News Recommendation: Google News Recommendation

The life cycle of news is very short, which means there are less useful ratings. So, how to recommend by such a few ratings in a short time? Google news is an online information portal site. It gathers thousands of news sources (after grouping the similar news) and displays them to users in a personalized manner.

Key idea: For the system which was published by Google news [25] in 2007, Google news employ a mixed algorithm of memory based and model based to generate recommendations. In 2010, Google news develops a content-based click pattern using a Bayesian framework [26], which can predict a user's current interest according to his own behavior and that of other users in the same region.

Details of Google news (2007): As the quantity of articles and users are very large and the expected response time is limited, pure memory-based recommendation is not applicable. Hence, Google uses a combination of model-based and memory-based technology in its system. The model-based part relies on two clustering techniques: probabilistic latent semantic indexing (PLSI) [27] and MinHash[6]. The basic idea of PLSI is similar to probabilistic clustering, which identify like-minded users and related articles, and cluster them together. Minhash puts the candidate objects (browsed by two users) in the same hash bucket. In addition, the approach of memory-based part is named "Adjoint PageView". It refers to an article browsed by the same user in a pre-defined period of time. Each of these method assigns a numeric score to an article, then the recommending scores R of the article a are deciding by Formula 3. In this formula, w_s is the weight given to the algorithm s, and r_a^s represents the score to article a given by algorithm s.

$$R = \sum w_s r_a^s \tag{3}$$

Details of Google news (2010): In the system released in 2010, they combined their existing system with an extra part, content-based recommendation. In this part, user's interest is divided into two parts: the interest of a user himself and the interest influenced by local news. The key method of generating forecast is as follows: At first, Google news uses all user's click history in different periods to predict user's real interests. Then, it integrates these predicted results together to obtain a more accurate results of user's real interest. Finally, it uses the user's real interest and the trend of local news to predict the user's current interests. Using $CR(s)$ represent the score calculated by the part of content-based algorithm for a candidate article, and $CF(s)$ the score calculated by Formula 3. These two scores are combined for new recommendation using Formula 4.

$$R = CR(s) \times CF(s) \tag{4}$$

Parallel approach: Google news uses its own MapReduce technology to distribute computing tasks among several clusters. MapReduce is a tool for parallel

[6] https://en.wikipedia.org/wiki/MinHash.

operation of large data sets, which is implemented with C++ programming language. Its main function is to provide a simple and powerful interface, to make computation be concurrent and be executed distributed automatically. MapReduce resolved the problems of calculating and obtaining the specified data from these massive data quickly.

Comparing to other recommendations, most of news only accumulates a few feedback, on account of its short lifecycle. It is necessary to find valuable information in a very limited time, and then recommend it to proper readers who may be interested in.

3.4 Movie Recommendation: Netflix (2012)

Compared to the news, movies accumulate much more feedback. But if the recommendation system cannot predict users preference accurately, it is likely to recommend films that do not meet the users taste, resulting in customer churn. As a successful online movie rental provider, Netflix[7] predict a users preference accurately. Meanwhile, it ensures timely updating of recommendation lists.

Key idea: On March 27, 2012, two engineers of Netflix published an article[8] in their official blog to introduce the architecture of Netflix. The system is consist of three parts: off-line, on-line and near-line. Figure 2 shows a screen capture from the original article. Soon after that, they released another article in their technology blog[9], providing more details about their ranking model.

Details of architecture: Online computation is expected to be more responsive to recent events and user interaction. Meanwhile, it must be done timely. These limit the amount of processed data and complexity of algorithms, and it may not meet Service-Level Agreement (SLA) in a certain type of situation. For the off-line computation, it has less restriction for data volume, algorithm complexity, and less requirement of time, but the data of off-line model obsolete easily. Nearline computation is a combination of these two models. Its performance is similar to online calculation, but doesn't need to complete in real time and the results are temporarily stored together. These make it be asynchronous and with faster response. Nearline approach utilizes the flow calculation to get some intermediate, which can either be sent to the online part to update the real-time recommendation model, or be stored for backups.

Details of ranking: The sorting part is done by off-line calculation. Instead of using a single model, they select, train and test lots of machine learning approaches. They keep tracks of multiple dimensions of indexes when testing, especially the residence time and the time of user's video playback. Generally, a plurality of A/B tests can be run in parallel, so as to verify multiple methods simultaneously. They put 6 different algorithms into A/B test weekly, and assess

7 https://www.netflix.com/.
8 http://techblog.netflix.com/2013/03/system-architectures-for.html.
9 http://techblog.netflix.com/2012/06/netflix-recommendations-beyond-5-stars.html.

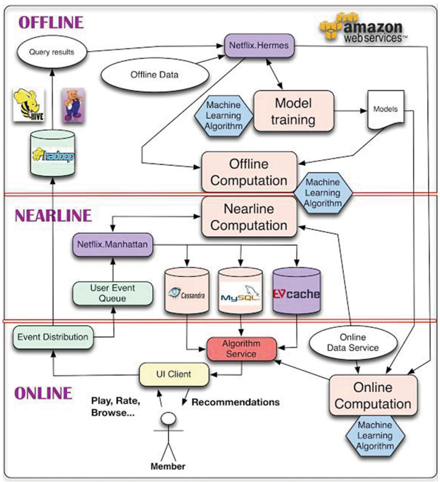

Fig. 2. The architecture of Netflix

offline and online indexes of these algorithms continuously. Then, the algorithm with excellent performance would be part of algorithm in their recommendation system.

Parallel approach: Amazon provides Hadoop PaaS platform to Netflix, and provides services to them through Elastic MapReduce (EMR). EMR provides apis and Hadoop cluster, where they can get one or more Hadoop jobs. Netflix has achieved Hadoop PaaS services (which is Genie (https://github.com/Netflix/gen-ie/wiki)). Genie could support thousands of concurrent jobs submit at the same time.

Summary: The above introduction of Google news and Netflix are the recommendation algorithms published before. Since they haven't disclosed their current system structures and main algorithms, we can only get the content published before, through either user interface or user experience speculate their operation.

4 Comparisons and Analyses

In this section, we analyze the mentioned systems above from seven aspects: the prediction accuracy, coverage, diversity, time complexity, cold-start, sparseness and personalization, as shown in Fig. 3. We can see that the performance of recommendation has been significantly improved during these years. It is worth noting that, in the following part, we will only compare the systems which have published the details on the specific aspects.

Accuracy: Accuracy is a measure of the ability of a recommendation system/algorithm on predicting users' behavior. Generally, it can be calculated from

	Accuracy	Coverage	Diversity	Instant-aneity	Cold-start	Sparsen-ess	Personaliz-ation
Google news(2007)	★★	★	★★	★★	★	★	E
Google news(2009)	★★★	★★	★★	★★	★★	★★	E
Netflix(2012)	★★★	★★★	★★	★★	★★	★★	E
Meituan(2015)	★★★★	★★★★	★★★★	★★★	★★★	★★★★	P
Facebook (2015)	★★★★	★★★★	★★★★	★★★★	★★★	★★★	P

Fig. 3. Comparison of systems (These comparision use the traditional recommendation system (each index are a star) as a benchmark for comparison. N, E, P in the column of "personality" represents non-personalization (we do not introduce non-personalization system in our paper), ephemeral personalization and persistent personalization respectively.)

the overlapping ratio of user's click behavior and recommendation list given by offline algorithms. The greater the overlapping ratio is, the higher the system accuracy is. The main purpose of recommendation system in real applications is to improve the prediction accuracy and obtain greater benefits. Therefore, each application uses a variety of methods to improve their accuracy. Among them, Meituan (2015) fuses data with multiple strategies, which improve prediction accuracy and thus ensure the accuracy of recommendation.

Coverage: The coverage metric describes the ability of a recommendation system to explore long tail goods [28] in a mass of goods. Which is generally defined as a ratio of recommended items to the total set of items. A Matthew effect is a developmental psychology phrase, it means the stronger are getting stronger and the small and weak are getting weaker. Google news (2007) can recommend a piece of news if and only if it has been clicked, there is a strong Matthew effect. In addition, it doesn't take the differences between users into account. Google news (2009) addresses those drawbacks by incorporating content-based method. In this way, the Matthew effect is reduced, and the coverage is increased. Query-based policy of Meituan (2015) helps to enhance the system's performance of exploring the long tail of goods and thus improves the coverage.

Diversity and noverty: The diversity metric describes the ability of a recommendation system in providing various recommendations, i.e., whether the recommendation results could override different interests of a specific user. Novelty means that the recommendation algorithm provides some novel items to a user, which he hasn't heard before and may not be similar to his historical records. Meituan (2015) takes multiple aspects into account and analyzes users' different interests comprehensively. These two policies both improve the diversity of the recommendation results. But most methods of Meituan (2015) are based on users' historical information, which leads to a low novelty.

Parallel and time complexity: It is worth noting that most of recommendation systems mentioned above are recommend by a variety of algorithms in parallel, improving the efficiency while ensuring the timeliness of the recommendation. Facebook (2015) uses multi-GPU training parallel framework and combining data parallel and model parallel to train models. Moreover, Facebook (2015) used iterative algorithms, SDG and ALS, to reduce the time complexity while work-to-work method could reduce the communication time effectively.

Cold-start: In recommendation systems, it is hard to make proper recommendations for new users, new items, and new systems, because there are none or very few ratings related to them. This leads to the cold-start problem. Google news (2009) introduces content-based method, making its results are not just rely on users' clicks. Meituan (2015) explores location-based policy to recommend items around users' location. These strategies can alleviate cold-start, to some extent.

Sparseness: Due to the fact that many users usually rate only a few items, most elements in the user-item matrix are zero. This phenomenon is called data sparsity. Location-based and query-based methods do not depend on the similarity matrix. Therefore, the impact brought by sparseness is limited in Meituan (2015). Furthermore, graph-based recommendation of Meituan (2015) uses transmissibility of similarity to obtain similar matrix more accurately and conveniently, which reduces the interference caused by sparseness.

Personalization: There are three categories about the degree of personalization [16], non-personalization, ephemeral personalization and persistent personalization. Non-personalization means that the recommendation results provided to each customer is identical. Ephemeral personalization is just simply making recommendations based on current browsing products or goods in a user's shopping cart. Persistent personalization means that the recommendations offered to different users are different even when they are looking at the same items.

5 Research Directions

In recent years, the developments of machine learning, large-scale network applications and high-performance computing have promoted recommendation system to a new upsurge. According to recent researches, particularly the achievements in ACM RecSys held in 2014[10] and 2015[11], we summarize some hotspots of recommendation system.

5.1 Hotspots for Long-Term

Context-aware recommendation systems (CARS): The utilization of context information alleviates sparsity and the decrease of recommendation accuracy caused by environmental changes, makes the system more intelligent and humanized. Hariri et al. [29] proposed an interactive recommendation system,

[10] http://recsys.acm.org/recsys14/.
[11] http://recsys.acm.org/recsys15/.

which uses the latest user behavior information to reconstruct model and to make recommendations, once the significant changes in the scene. Jiang et al. [30] propose a time-evolving rating prediction scheme in trust-based recommendation systems, using fluid dynamic theory.

Hybrid system: Hybrid a variety of recommendation methods makes the system access to their respective strengthes. For instance, the combination of collaborative filtering algorithm (CF) and content-based [31] or MP (Most Popular) ease cold start problem. Combining CF with social networks based method [32] alleviate problems caused by data sparseness.

Security issues: Security issues include the securities of system and user privacy. Seminario et al. [33] presents a strong program attack model (PIA) from the perspective of an attacker. It proves that attackers could attack collaborative recommendation systems based on SVD, user-based and item-based. Meanwhile, the experimental results of Frey et al. [34] show that in the case of CF, although the attacker forge a false identity to carry out attacks, the inherent similarities between real users protect the interests of users to some extent.

Social network: Social network covers all forms of network services around human society, and now becomes an indispensable part of human life. In fact, social networks are overlapping [35]. Making use of the overlapped identities of an user in different social networks can alleviate cold start problem and data sparseness problem. Furthermore, social networks are reflections of real life. People are more likely to be influenced by recommendation from friends. Jiang et al. [36] present the idea to evaluate trust by selecting proper recommenders.

In these research focus, security is a constant topic in the future for a long period, while both hybrid recommendation and CARS will be hot topics, until the "new darling" of the recommendation system appears.

5.2 Open Challenges

Cold-start: Cold-start problem exists even from the very beginning of recommendation system. There are many ways to partially solve the problem, but it is difficult to settle it. For instance, Ji and Shen [37] proposes a novel method to alleviate cold-start problem. They first build tag-keywords relation matrix based on the statistics, then select tags and extract keywords by a 3-factor matrix factorization model, and integrate the vectors at last. However, if a user has no record, the cold-start problem still exists.

Diversity and novelty: As we have mentioned in Sect. 4, many recommendation systems have a high prediction accuracy, but low variety and novelty. Vargas et al. [38] use backwards thinking to find users for items, which improves the diversity of recommendation successfully. However, there is still no effective way to ensure the novelty of recommended results while keeping high accuracy.

The directions above have troubled researchers for a long time, and these will persecute researchers in the future. Among them, diversity and novelty are contradict with accuracy in some degree. How to find a proper balance between them will also be the future research focus.

5.3 Meaningful Directions

Parallelization: As the volume of data growing, the effective integration of recommendation system and high-performance computing is becoming inevitable, for example, implementing the recommendation system [39] and processing data [40] on Hadoop. Well combination of recommendation system and high-performance computing can overwhelmingly improves computing performance and reduces computing time.

Interface display: The way of presenting recommendation results to users would affect their first impression on the system. Vanchinathan et al. [41] take advantage of the similarities between users or items to solve this problem. In addition, how to explain the recommendation result is also very important. Users often take a skeptical attitude towards recommendation results, so reasonable interpretation makes results more convincing.

How to do more tasks more efficiently? How to incorporate it into recommendation system? How to make users have more trust in our recommendation and take our advice? All the above aspects are worth further studies.

6 Conclusion

Recommendation system has been an effective tool to alleviate information overload. However, current recommendation systems still need to be improved to make the recommendation methods more effective in a broader range of applications, and make the results more in line with users' interests and needs.

In this paper, we introduce a range of representative recommendation applications and analyze the improvements in different periods. Based on this, we summarize the research focuses and open challenges as well as significant research directions. We also review the developments of the latest researches. Our work tries to provide some insights on future researches. In the real world, more factors should be considered than we have mentioned. Portability, scalability, robustness, and the ability to handle large data are issues we will take into account in future work. We believe that academic research should be able to guide the design of practical applications, that is why we choose to survey the real applications. We hope these issues we proposed in our paper can help to promote the developments of future applications of recommendation systems.

Acknowledgments. This work is supported by NSFC grants 61502161, 61472451, 61272151, the Chinese Fundamental Research Funds for the Central Universities 531107040845, and the National High-tech R&D Program of China 2014AA01A302 and 2015AA-015305.

References

1. Goldberg, D., Nichols, D., Oki, B.M., Terry, D.: Using collaborative filtering to weave an information tapestry. Commun. ACM **35**(12), 61–70 (1992)

2. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: CSCW, pp. 175–186. ACM (1994)
3. Liu, L., Lecue, F., Mehandjiev, N.: Semantic content-based recommendation of software services using context. TWEB **7**(3), 17–36 (2013)
4. Di Noia, T., Mirizzi, R., Ostuni, V.C., Romito, D., Zanker, M.: Linked open data to support content-based recommender systems. In: I-SEMANTICS, pp. 1–8. ACM (2012)
5. Jung, G., Mukherjee, T., Kunde, S., Kim, H., Sharma, N., Goetz, F.: Cloudadvisor: a recommendation-as-a-service platform for cloud configuration and pricing. In: SERVICES, pp. 456–463. IEEE (2013)
6. Carrer-Neto, W., Hernández-Alcaraz, M.L., Valencia-García, R., García-Sánchez, F.: Social knowledge-based recommender system. Application to the movies domain. Expert Syst. Appl. **39**(12), 10990–11000 (2012)
7. Park, Y., Park, S., Jung, W., Lee, S.: Reversed CF: a fast collaborative filtering algorithm using a k-nearest neighbor graph. Expert Syst. Appl. **42**(8), 4022–4028 (2015)
8. Jiang, S., Qian, X., Shen, J., Fu, Y., Mei, T.: Author topic model based collaborative filtering for personalized POI recommendation. TMM **6**, 907–918 (2015)
9. Zhou, Y., Wilkinson, D., Schreiber, R., Pan, R.: Large-scale parallel collaborative filtering for the Netflix prize. In: Fleischer, R., Xu, J. (eds.) AAIM 2008. LNCS, vol. 5034, pp. 337–348. Springer, Heidelberg (2008). doi:10.1007/978-3-540-68880-8_32
10. Majid, A., Chen, L., Chen, G., Mirza, H.T., Hussain, I., Woodward, J.: A context-aware personalized travel recommendation system based on geotagged social media data mining. IJGIS **27**(4), 662–684 (2013)
11. Jamali, M., Ester, M., Trustwalker: a random walk model for combining trust-based and item-based recommendation. In: SIGKDD, pp. 397–406. ACM (2009)
12. Yin, H., Cui, B., Chen, L., Zhiting, H., Zhang, C.: Modeling location-based user rating profiles for personalized recommendation. TKDD **9**(3), 19 (2015)
13. Zhang, Y., Zhang, M., Zhang, Y., Lai, G., Liu, Y., Zhang, H., Ma, S.: Daily-aware personalized recommendation based on feature-level time series analysis. In: WWW, pp. 1373–1383. ACM (2015)
14. Debnath, S., Ganguly, N., Mitra, P.: Feature weighting in content based recommendation system using social network analysis. In: WWW, pp. 1041–1042. ACM (2008)
15. Ben Schafer, J., Konstan, J., Riedl, J.: Recommender systems in e-commerce. In: EC 1999, pp. 158–166. ACM (1999)
16. Ben Schafer, J., Konstan, J.A., Riedl, J.: E-commerce recommendation applications. In: Applications of Data Mining to Electronic Commerce, pp. 115–153. Springer (2001)
17. Bao, J., Zheng, Y., Wilkie, D., Mokbel, M.F.: A survey on recommendations in location-based social networks. GeoInformatica **19**(3), 525–565 (2014)
18. Jiang, W., Wang, G., Alam Bhuiyan, M., Wu, J.: Understanding graph-based trust evaluation in online social networks: methodologies and challenges. ACM Comput. Surv.**49**(1) (2016). Article 10
19. Karydi, E., Margaritis, K.G.: Parallel and distributed collaborative filtering: a survey. arXiv preprint arXiv:1409.2762 (2014)
20. Liang, H., Hogan, J., Yue, X.: Parallel user profiling based on folksonomy for large scaled recommender systems: an implimentation of cascading MapReduce. In: ICDMW, pp. 154–161. IEEE (2010)

21. Christou, I.T., Amolochitis, E., Tan, Z.-H.: Amore: design and implementation of a commercial-strength parallel hybrid movie recommendation engine. Knowl. Inf. Syst. **47**, 1–26 (2015)
22. Herbert, R., Monro, S.: A stochastic approximation method. Ann. Math. Stat. **22**(3), 400–407 (1951)
23. Jack, K., Wolfowitz, J.: Stochastic estimation of the maximum of a regression function. Ann. Math. Stat. **23**, 462–466 (1952)
24. Volinsky, C., Koren, Y., Bell, R.: Matrix factorization techniques for recommender systems. Computer **42**, 30–37 (2009)
25. Das, A.S., Datar, M., Garg, A., Rajaram, S.: Google news personalization: scalable online collaborative filtering. In: WWW, pp. 271–280. ACM (2007)
26. Liu, J., Dolan, P., Pedersen, E.: Personalized news recommendation based on click behavior. In: IUI, pp. 31–40. ACM (2010)
27. Hofmann, T.: Probabilistic latent semantic indexing. In: SIGIR, pp. 50–57. ACM (1999)
28. Anderson, C.: The long tail: why the future of business is selling more for less, Hyperion (2006)
29. Hariri, N., Mobasher, B., Burke, R.: Context adaptation in interactive recommender systems. In: ACM RecSys, pp. 41–48. ACM (2014)
30. Jiang, W., Wu, J., Wang, G., Zheng, H.: Forming opinions via trusted friends: time-evolving rating prediction using fluid dynamics. IEEE Trans. Comput. (2015). doi:10.1109/TC.2015.2444842
31. Saveski, M., Mantrach, A.: Item cold-start recommendations: learning local collective embeddings. In: ACM RecSys, pp. 89–96. ACM (2014)
32. Sedhain, S., Sanner, S., Braziunas, D., Xie, L., Christensen, J.: Social collaborative filtering for cold-start recommendations. In: ACM RecSys, pp. 345–348. ACM (2014)
33. Seminario, C.E., Wilson, D.C.: Attacking item-based recommender systems with power items. In: ACM RecSys, pp. 57–64. ACM (2014)
34. Frey, D., Guerraoui, R., Kermarrec, A.-M., Rault, A.: Collaborative filtering under a sybil attack: analysis of a privacy threat. In: EuroSec, p. 5. ACM (2015)
35. Rossi, L., Magnani, M.: The ML-model for multi-layer social networks. In: ASONAM, pp. 5–12. IEEE (2011)
36. Jiang, W., Wu, J., Wang, G.: On selecting recommenders for trust evaluation in online social networks. ACM Trans. Internet Technol. (TOIT) **15**(4) (2015). Article 14
37. Ji, K., Shen, H.: Addressing cold-start: scalable recommendation with tags and keywords. Knowl. Based Syst. **83**, 42–50 (2015)
38. Vargas, S., Castells, P.: Improving sales diversity by recommending users to items. In: ACM RecSys, pp. 145–152. ACM (2014)
39. Meng, S., Dou, W., Zhang, X., Chen, J.: Kasr: a keyword-aware service recommendation method on mapreduce for big data applications. TPDS **25**(12), 3221–3231 (2014)
40. Wang, C., Zheng, Z., Yang, Z.: The research of recommendation system based on Hadoop cloud platform. In: ICCSE, pp. 193–196. IEEE (2014)
41. Vanchinathan, H.P., Nikolic, I., De Bona, F., Krause, A.: Explore-exploit in top-n recommender systems via Gaussian processes. In: ACM RecSys, pp. 225–232. ACM (2014)