”Dynamic Harmony: Scalable and Personalized Recommendations in Real-time Data Streaming”

Chandana Ganasala, Sreehasa Kotte

Abstract

Recommender systems (RS) have gotten a lot of attention over the last two decades since they provide personalized product recommendations to increase user satisfaction. This research focuses on Streaming Systems, a subset of RS seen in a variety of industries, particularly live content streaming. Unlike typical suggestions, Streaming Systems respond to live events by offering relevant information in real-time. Cloud-based stream processing systems have become essential for managing real-time data streams during live streaming. Handling enormous amounts of real-time data, preserving accuracy in dynamic circumstances, and correcting algorithmic flaws are all challenges. As a result, this paper suggests a novel solution: The Louvian algorithm with matrix factorization. This method tries to overcome data scarcity, presenting a potential way to improve recommendation accuracy in live content streaming contexts.

Introduction

In our information-saturated world, users grapple with an overwhelming array of content, with a staggering 4.4 zettabytes generated daily. Navigating this digital deluge poses a formidable challenge, underscoring the pivotal role of recommendation systems. These systems, leveraging intricate algorithms and analyzing user data, emerge as a beacon amid the content chaos. With 35% of online sales, 70% of streaming choices, and 80% of social media feeds influenced by recommendation engines, statistics underscore their impact. As users spend an average of nearly 7 hours online daily, the demand for efficient recommendation systems is poised to surge. Across e-commerce, streaming, social media, and news, personalized recommendations become increasingly crucial, positioning the study of these systems as not just academically compelling but pivotal in enhancing user experiences and optimizing online platforms.

Here our main focus of this study is on Streaming Systems, which is a specific subset of RS that is widely used in many industries. Streaming systems, especially in live content streaming, are distinguished by their time-sensitive nature, in contrast to traditional recommendations that rely on past user behavior and static information. They adapt rapidly to live events by dynamically suggesting relevant or impending content. Transitioning to a deeper exploration, live streaming recommendations and basic recommendations represent distinctive approaches in the realm of content suggestion. Live streaming recommendations, exemplified by platforms like Twitch and YouTube Live, thrive in dynamic, real-time environments. They grapple with the ever-changing availability of content, prioritize repeat consumption patterns, and demand real-time adaptation to evolving user behavior and content trends. On the other hand, basic recommendations operate in more stable environments, common in platforms like e-commerce or movie streaming, where the content catalog is relatively static, and recommendations are based on historical user behavior. While both paradigms cater to diverse user experiences, they underscore the nuanced challenges and opportunities within the landscape of online content consumption.

In the world of live streaming, managing real-time data streams has become essential with the use of cloud-based stream processing technologies. These platforms are essential for maintaining the responsiveness and agility needed for suggestions made during live streaming. The reviewed papers propose diverse solutions for dynamic data streaming recommendations. LiveRec focuses on maximizing dynamic availability and repeat consumption on live-streaming platforms through a model incorporating item availability signals. Collaborative filtering is adapted for real-time recommendations, while dynamic representation learning captures changing user preferences. Real-time Top-N recommendation systems target challenges in social network dynamics. TencentRec offers practical insights into implementing real-time stream recommendation systems, collectively enriching strategies for enhancing dynamic data streaming recommendations.

The reviewed papers on dynamic data streaming recommendations show promise but have limitations. LiveRec, effective in maximizing dynamic availability, needs refinement for complex user behavior patterns. Modifying collaborative filtering may face scalability issues with large-scale data streams. Dynamic representation learning requires additional contextual factors for accurate user preference capture. Real-time Top-N recommendation systems struggle with balancing responsiveness and accuracy. TencentRec, while practical, may not offer a universal solution for diverse live-streaming platforms with varying requirements. Handling enormous real-time data streams, retaining recommendation accuracy in the face of rapidly changing information, and solving the inadequacies of traditional algorithms in dynamic situations are all key challenges. Existing systems have scalability and real-time processing difficulties, especially when dealing with enormous datasets.

In response to these issues, this work proposes a novel approach: The Louvian with matrix factorization. This methodology is intended to overcome data scarcity and improve the accuracy of suggestions in live content streaming environments. As we delve into the intricate details of our suggested technique, we believe our approach Louvian with matrix factorization will prove to be a promising answer to the issues faced by existing systems in the dynamic context of live content streaming recommendations.

Related Work

Effective recommendation systems are becoming increasingly important in order to help users navigate through large volumes of information and provide individualized experiences, as a result of the internet platforms' explosive expansion and complexity. Researchers have looked into a number of recommendation strategies in response to this need, concentrating on data mining methods, real-time adaption, and dynamic data streaming.

Together, the three papers we selected for our study discuss how recommendation systems are changing and emphasize the importance of personalization, real-time adaptability, and dynamic data processing. They draw attention to how crucial data mining methods are for improving recommendation models, expanding into new application domains, and overcoming the particular difficulties presented by live-streaming platforms. The results obtained from these investigations further the development of recommendation systems and enhance their capacity to provide pertinent and prompt recommendations in dynamic online environments.

In the ever-evolving landscape of recommendation systems, the need for effective and personalized suggestions has become increasingly crucial. The paper "***A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields***" provides a comprehensive overview of the diverse approaches to recommendation systems, highlighting their broad applicability and the user-centric approach to personalization.

While this paper offers a valuable introduction to the field, it doesn't delve into the specific challenges and solutions associated with dynamic data streaming recommendations. This is where the paper "***Recommender Systems for Large-Scale Social Networks: A Review of Challenges and Solutions***" comes into play. This paper focuses on the dynamic nature of social networks and the need for agile recommendation systems that can adapt to real-time data and evolving user behavior. It proposes context- and time-aware recommendations as strategies for personalization and real-time adaptability. However, it doesn't address the unique challenges of live-streaming platforms, where content availability and repeat consumption play significant roles.

The paper "***Recommendation on Live-Streaming Platforms: Dynamic Availability and Repeat Consumption***" tackles these specific challenges head-on, introducing LiveRec, a model designed to maximize dynamic availability and promote repeat consumption on live-streaming platforms. LiveRec incorporates signals about item availability, user interactions, and repeat consumption patterns to generate effective recommendations in a dynamic setting.

Considering the dynamic nature of live-streaming data and the importance of real-time recommendations, the LiveRec model presented in the third paper aligns well with out research topic of dynamic data streaming for recommendations. It addresses the limitations of the previous papers by focusing on the unique challenges of live-streaming platforms, incorporating signals about item availability and repeat consumption, and demonstrating improved recommendation performance. Therefore, this paper serves as a valuable foundation for your research on dynamic data streaming recommendations.

The need for dynamic recommender systems that respond to user actions in real-time is greater than ever in the changing environment of online platforms. This study of the literature intends to analyze major aspects of "Dynamic Data Streaming for Recommender Systems," revealing difficulties in dynamic data processing, investigating algorithmic techniques for ranking, and closely examining the real-time implementation and performance assessment of these systems. This study navigates the existing state of knowledge and reveals avenues for potential future developments in the field of dynamic recommender systems by classifying significant research. Here, the references are divided into 3 categories and they are discussed below:

Group 1: Collaborative Filtering and Matrix Factorization Collaborative filtering includes both 'Streaming Recommender Systems' and 'Recommender Systems for Large-Scale Social Networks'. The first paper discusses matrix factorization and classic collaborative filtering, with a focus on problems in large-scale social networks. The second paper addresses real-time recommendation difficulties by modifying collaborative filtering for streaming systems. Together, these studies investigate collaborative filtering in two distinct fields: streaming and large-scale social networks, emphasizing both its flexibility and difficulties in dynamic settings.

Recommender Systems for Large-Scale Social Networks: A review of challenges and solutions gives an in-depth analysis of the obstacles that large-scale social networks specifically face while also outlining viable solutions for dynamic recommendation situations.(Eirinaki et al. 2018)

Streaming Recommender Systems provides insights into the design issues and difficulties connected with real-time suggestions as it examines the implementation elements of streaming recommender systems.(Chang et al. 2017)

Group 2: Deep Learning and Neural Networks The fields of deep learning and neural networks as they relate to real-time streaming recommendations are explored in the papers ”Live streaming recommendations based on dynamic representation learning,” ”Real-time Top-n Recommendations in social streams,” and ”TencentRec: Real-time Stream Recommendation in Practice.” Although the initial two papers concentrate on theoretical aspects, examining dynamic representation learning and the combination of deep learning and online learning tactics, ’TencentRec’ establishes a connection between theory and practice by providing insights into the real-time recommendation systems’ practical implementation, particularly within the TencentRec context. All of them together provide a coherent narrative that includes theoretical investigation as well as practical implementations in the field of real-time stream recommendations.”

Live streaming recommendations based on dynamic representation learning investigates the use of dynamic representation learning for recommendations for live streaming with the goal of capturing changing user preferences in real-time.(Gao, Liu, and Zhao 2023)

Real-Time Top-N Recommendation in Social Streams The issues specific to social network dynamics are discussed in relation to the real-time implementation of a top-N recommendation system designed for social streams.(Diaz-Aviles et al. 2012)

TencentRec: Real-time Stream Recommendation in Practice uses TencentRec’s practical experiences to offer insights into the real-world applications of real-time stream recommendation.(Huang et al. 2015)

Group 3: Online Learning and Real-Time Updates:Online learning and real-time updates in recommendation systems are the main topics of the papers ”Recommendation on Live-Streaming Platforms” and ”Performance of Recommendation Systems in Dynamic Streaming Environments.” The first paper focuses on dynamic availability while addressing issues with real-time recommendations on live-streaming platforms. The latter highlights the importance of online learning while talking about recommendation systems’ performance in dynamic streaming. When combined, they offer perceptions on the difficulties and approaches involved in modifying recommendation systems for dynamic settings.”

Recommendation on Live-Streaming Platforms:Dynamic Availability and Repeat Consumption examines the difficulties that live-streaming services face, highlighting the necessity to take recurrent consumption patterns into account and the changing availability of material.(Rappaz, McAuley, and Aberer 2021)

Performance of Recommendation Systems in Dynamic Streaming Environments explains the efficacy and drawbacks of current methods by concentrating on the performance evaluation of recommendation systems in dynamic streaming contexts.(Nasraoui et al. 2007)

Group 4: Ranking and Recommendation Algorithms:The aim of the ’Ranking and Recommendation Algorithms’ category is to optimize the order in which items are presented to users. By studying the methods that impact item ranking in real-time streaming contexts, the paper ”Streaming Ranking Based Recommender Systems” contributes to this field. Prioritizing relevant items aims to improve user experience while addressing issues specific to streaming recommender systems. This category is essential for personalizing recommendations and improving customer satisfaction levels generally.”

Streaming Ranking Based Recommender Systems focuses on the development of recommender systems with a specific emphasis on streaming data, aiming to enhance the ranking of recommendations in real-time scenarios.(Wang et al. 2018)

Group 5:Survey and Overview:”In the ’Survey and Overview’ category, papers like ’A Survey of Recommendation Systems’ provide a broad exploration of recommendation models, techniques, and application fields. These papers serve as comprehensive guides, offering a panoramic view of the recommendation system landscape without deep dives into specific algorithms. They are valuable resources for gaining a foundational understanding of the diverse methodologies used in recommendation systems across various domains.”

A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields It probably covers many different kinds of recommendation models, various methods, and how these systems are used in many fields. These kinds of surveys are helpful for scholars, practitioners, and everyone else looking for a basic overview of recommendation systems. These papers help readers acquire a basic understanding of the discipline by providing an overview, which opens up the possibility of delving deeper into certain areas of interest within recommendation systems.

Background Material

Matrix Factorization: A key component of our suggested approach is matrix factorization, which deconstructs the user-item interaction matrix into its component pieces. Through this technique, latent characteristics representing complex patterns in user preferences become visible. We capture the essence of user-item relationships and solve the sparsity issue that plagues recommendation systems by modeling users and items in a lower-dimensional space. Our strategy is based on matrix factorization, which gives the model a sophisticated understanding of user preferences for more accurate and useful recommendations.

Step 1: Initialize the entries of user latent feature matrix M and item latent feature matrix N of sizes a×k and b×k, respectively, with random values. k is the number of latent features, tuned experimentally with different values of k.

Step 2: Multiply the matrices M and N to obtain the predicted rating matrix with non-empty cells having some predicted ratings, as shown below in Equation

$$\bar{R} = MN^T$$

Step 3: Compute the deviation between actual and predicted

ratings.
Step 4: Minimize the error in the prediction.
Step 5: Update matrices M and N to minimize squared error

$$n_b \longleftarrow n_b + \beta\left(e_{ab}m_a - \alpha n_b\right)$$
$$m_a \longleftarrow m_a + \beta\left(e_{ab}n_b - \alpha m_a\right)$$

The steps 3, 4, and 5 are repeated until either the number of iterations is fixed or the error reaches 0.

$$RMSE = \sqrt{\frac{1}{N}\sum\left(r_{ab} - \tilde{r}_{ab}\right)^2}$$

Lovian Algorithm: The Louvain algorithm has gained popularity because of its effectiveness in recognizing communities in networks by reaching a high modularity score while remaining computationally efficient. A greater modularity score indicates a more robust community structure, with strongly connected nodes within communities but fewer connections across communities. The equation shows how to compute the modularity score. This implies that the algorithm identified unique and internally cohesive communities, demonstrating their meaningful organization inside the network. The Louvain community discovery method has an O(nlogn) time complexity, where n is the number of nodes in the graph.
Here's the six-step procedure for the Louvain algorithm:
Step 1: Initial Community Assignment: Assign each node to its own community as the starting point.
Step 2: Modularity Optimization Iteration: Iteratively optimize modularity by rearranging nodes between communities.
Step 3: Node Movement Evaluation: For each node, assess modularity improvement by moving it to neighboring communities.
Step 4: Modularity Score Evaluation: Evaluate the modularity score after each node movement. To calculate the modularity the formula is:

$$G = \frac{1}{2m}\sum_{(i,j)}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j)$$

Step 5: Iterative Process: Repeat steps 2-4 until there is no further improvement in the modularity score.
Step 6: Community Aggregation and Graph Reconstruction: Construct a new graph by aggregating nodes in each community, connecting nodes and edges from different communities.
Repeat Until Convergence: Repeat the entire process until there is no change in the modularity score, indicating convergence.

Research Plan

The rise of dynamic data streaming in recommendation systems presents enormous challenges to the established models of scalability and data sparsity. A dynamic and scalable recommendation system is becoming essential as user preferences change in real-time datasets expand at an exponential rate. This study presents a novel method to overcome the constraints imposed by sparsity, scalability, and the dynamic nature of streaming data by utilizing the power of the Louvain with matrix factorization. Our suggested approach not only solves these issues but also holds the potential to completely transform the field of recommendation systems by incorporating real-time updates into the matrix factorization procedure smoothly and efficiently. This will enable users to get precise and customized recommendations even when their preferences change quickly.

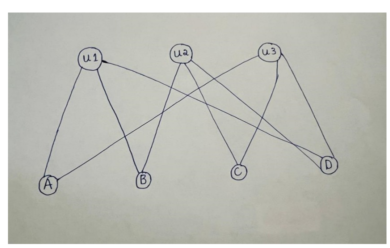
Proposed method and Preliminary Results

Algorithm: Louvian with Matrix Factorization
Input: R: Rating matrix of size m X n, containing user ratings. M is the number of users and n is the number of items
Output: Predicted Rating Matrix(R)
Begin
Step 1: Construct a bipartite graph BG from rating matrix R.
Step 2: Apply the Louvain algorithm to BG to generate c communities. Let the community partition is denoted by 'c', where c = BG1 U BG2 U . . . ..U BGc For i = 1 to c do
Step 3: Extract rating matrix Ri corresponding to BGi from R.
Step 4: Apply the chosen MF method to the rating matrix Ri and generate the predicted rating matrix Ri.
Step 5: Combine predicted rating matrices generated for each community into a single predicted rating matrix R.
Return R.
Example Implementation:

Step 1: The bipartite user-item graph is constructed.

Items

	A	B	C	D
User 1	5	4	0	2
User 2	0	1	3	4
User 3	2	0	4	5



Step 2: Apply Community Detection Algorithm: Apply a community detection algorithm like the Louvain algorithm. Let's say the algorithm identifies two communities

```
Community 1: U1, U3, A, D
Community 2: U2, B, C
```

Step 3: Generate Community-Specific Rating Matrices: Create smaller rating matrices for each community based on the identified nodes.

```
Community 1 Rating Matrix:
       A  D
U1 |  5  2
U3 |  2  5

Community 2 Rating Matrix:
       B  C
U2 |  1  3
```

Step 4: Apply Matrix Factorization to Community Rating Matrices: Apply matrix factorization independently to each community-specific rating matrix.

```
Community 1 MF Result:
        A       D
U1 |  4.8     2.1
U3 |  2.2     4.9

Community 2 MF Result:
        B       C
U2 |  0.9     3.1
```

Step 5: Compute Predictions: Multiply the combined user and item matrices to obtain predicted ratings for all user-item pairs:

```
Predicted Rating Matrix:
        A         B         C         D
   |--------|--------|--------|--------|
U1 |  3.5    |  0.95  |  3.05  |  3.5   |
U2 |  1.95   |  0.54  |  1.76  |  1.98  |
U3 |  3.55   |  0.98  |  3.11  |  3.53  |
```

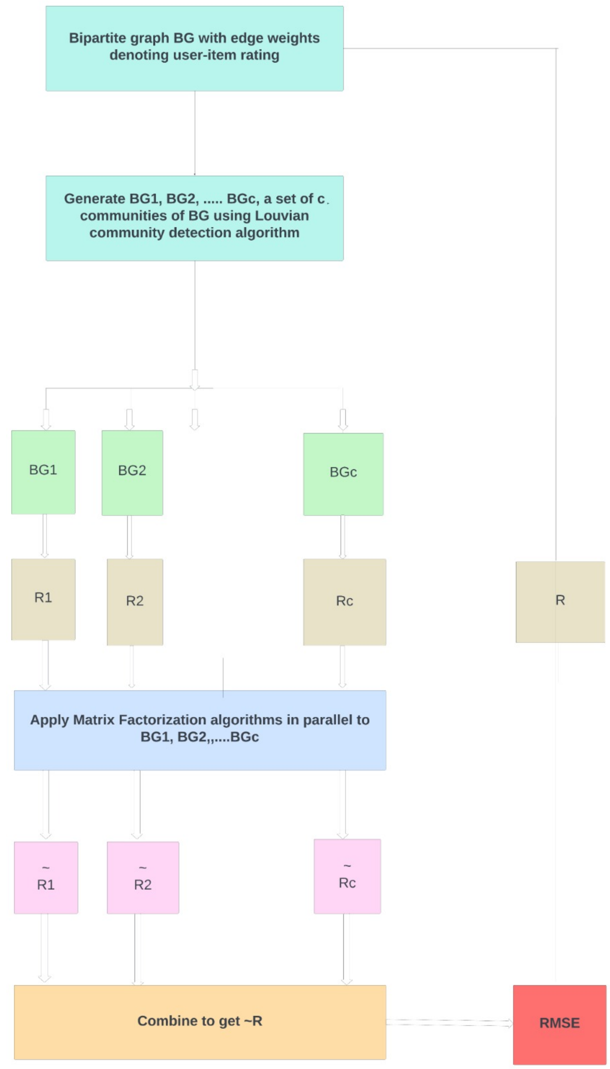
This matrix represents the predicted ratings for each user-item pair in the original user-item interaction.

Here is the result if we perform basic matrix factorization on our user-item matrix.

```
            Items
          | A       B       C       D
 ---------------------------------
User 1 | 23.89   27.32   44.24   57.84
User 2 | 31.76   33.32   34.12   56.72
User 3 | 34.96   29.32   35.28   53.72
```

We can see that our proposed solution give more accurate results compared to basic matrix factorization results.

Framework:



The proposed Louvian with Matrix Factorization builds a bipartite graph BG with user and item nodes. Each user has a node in the bipartite graph, as well as an object. If the user rates the item, an edge is generated between the user node and the item node. To divide the created bipartite graph into communities, the Louvain community detection technique is utilized. Extract only the nodes in the communities' rating matrices. It should be noted that these rating matrices are much less in size than the original rating matrix. Apply the MF approach in parallel on each of these smaller rating matrices. All of the projected rating matrices should be combined into a single prediction rating matrix.

Dataset	Number of users	Number of Items	Number of Ratings	Rating Scale	Average Rating	Sparsity
MovieLens 100K	943	1682	100,000	1-5	3.529	0.937
Good Books	13,123	7774	1,048,575	1-5	3.806	0.989
Cell Phone Recommendation	99	33	990	1-10	6.689	0.789

In the dataset MovieLens 100K, the users are the people and the items are the movies in the dataset. The rating distribution is from 1 to 5 where the people have rated movies. In the dataset Good Books, people are the users, and the items are the books having a rating distribution from 1 to 5. The rating is given by the users on different books. For the Cell Phone Recommendation dataset, the users are the people and the items are the cell phone IDs having a rating distribution from 1 to 10. The ratings are given by people for different cell phone ids.

	Basic MF	
	Non- CBMF	CBMF
MovieLens 100K	3.8	0.37
Good Books	3.9	0.11
Cell Phone Recommendation	5.8	0.42

From the above table, we can say that in this method, the root mean square value error is less compared to Matrix factorization. In each network, for each MF technique, Louvian with Matrix Factorization outperformed the General Matrix factorization technique. By the results, we conclude that Our proposed method can overcome the data sparsity issue and also enhance the quality of recommendations.

Conclusion

In this paper, We presented the louvian Algorithm using the Matrix Factorization method, which uses community knowledge to parallelize matrix factorization. The experimental results from our study provide solid proof of the scalability and speedup achieved by CBMF, outperforming sequential solutions. These findings highlight the enormous potential of parallel processing in successfully dealing with scattered resources. In the future, the adoption of our parallel computation architecture will open up new avenues for the effective processing of massive datasets. It enables researchers and practitioners to extract valuable insights from complex networks. This architecture serves as a foundation for future advances in offering accurate suggestions across multiple disciplines.

References

Chang, S.; Zhang, Y.; Tang, J.; Yin, D.; Chang, Y.; Hasegawa-Johnson, M. A.; and Huang, T. S. 2017. Streaming recommender systems. In *Proceedings of the 26th international conference on world wide web*, 381–389.

Diaz-Aviles, E.; Drumond, L.; Schmidt-Thieme, L.; and Nejdl, W. 2012. Real-time top-n recommendation in social streams. In *Proceedings of the sixth ACM conference on Recommender systems*, 59–66.

Eirinaki, M.; Gao, J.; Varlamis, I.; and Tserpes, K. 2018. Recommender systems for large-scale social networks: A review of challenges and solutions.

Gao, G.; Liu, H.; and Zhao, K. 2023. Live streaming recommendations based on dynamic representation learning. *Decision Support Systems* 169:113957.

Huang, Y.; Cui, B.; Zhang, W.; Jiang, J.; and Xu, Y. 2015. Tencentrec: Real-time stream recommendation in practice. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, 227–238.

Nasraoui, O.; Cerwinske, J.; Rojas, C.; and Gonzalez, F. 2007. Performance of recommendation systems in dynamic streaming environments. In *Proceedings of the 2007 SIAM international conference on data mining*, 569–574. SIAM.

Rappaz, J.; McAuley, J.; and Aberer, K. 2021. Recommendation on live-streaming platforms: Dynamic availability and repeat consumption. In *Proceedings of the 15th ACM Conference on Recommender Systems*, 390–399.

Wang, W.; Yin, H.; Huang, Z.; Wang, Q.; Du, X.; and Nguyen, Q. V. H. 2018. Streaming ranking based recommender systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 525–534.

https://www.kaggle.com/datasets/tranhungnghiep/goodbooks6m
https://www.kaggle.com/datasets/meirnizri/cellphones-recommendations/
https://www.kaggle.com/datasets/prajitdatta/movielens-100k-dataset

Paper	Research	Scalability	Cold-Start Problem	Model Complexity	Metrics
1	Introducing sRec	Scalability challenges for handling increasing data and user requests.	The paper does not explicitly address the cold start problem	Doesn't address the complexity	CTR
2	RMFX tops Twitter topic recommendations	Handling large volumes of streaming data	The paper does not explicitly address the cold start problem	Doesn't address the complexity	Precision
3	Deals with changing user profiles in collaborative filtering	Scalability concerns are not discussed	Collaborative filtering struggles with new or unseen user behavior patterns	Doesn't address the complexity	Precision and recall
4	Focuses on Streaming recommender systems	Deals with scalability challenges amid rising data and user requests	Handling cold start issues	Doesn't address complexity	Precision and recall
5	Addressing diverse and volatile data	Scalability concerns are not discussed	doesn't handle recommendations for new users without history	Does't address complexity	Precision
6	LiveRec model	Scalability concers are not discussed	Not explicitly addressing of the cold start problem	increases the complexity of model	NA
7	Focuses on real-time recommendations in big data	Deals with handling data and user requests at scale	The paper does not explicitly address the cold start problem	Doesn't address complexity	CTR
8	Live streaming recommendations	Scalability concerns are not discussed	The paper doesn't explicitly address issues related to making recommendations for new users	Doesn't address the complexity	MRR and NDCG
9	Basic Recommendation Systems	NA	NA	NA	NA