Vaishali Surjuse, 2021, 9:4
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

Scalable Personalized Content Recommendation Based on a Modular Approach

Prof. Vaishali Surjuse, Vipul Petkar, Mamta Balbudhe, Huzefa Motorwala, Harshad Nimsarkar
Department of Computer Technology
KDK college of engineering
Nagpur, Maharashtra, India.
vsurjuse22.kdkce@gmail.com, Petkar.vipul@gmail.com, mamtabalbudhe333@gmail.com, huzefamotorwala5225@gmail.com, harshadnimsarkar@gmail.com

Abstract- Traditional recommendation methods have failed to capture complex interaction and behavior of users whereas purely deep learning models are very expensive to deploy at scale. In this paper, we describe a system at a high level consisting of two parts: first, we detail an efficient generation model with caching and then a separate ranking model that delivers precise results. This modular approach allows fine-tuning of every component and increases user engagement.

Keywords:- Recommender system; Collaborative Filtering; Personalized Recommendations;

I. INTRODUCTION

With the ever-growing volume of online content produced, it is getting increasingly difficult for people to find the good content they want to watch. A recommendation engine learns from user preferences and helps deliver personalized results in return maximizing user engagement. Thus, having an effective recommendation system is necessary.

The three major challenges faced while Building a good recommendation system can be

1. Scale:
Many recommendation algorithms proven to have worked well on small sets but fail to operate on a large corpus of content mostly due to the sparsity of data.

2. Cost:
Modernend-to-end approaches to recommendations such as Deep Neural Networks and autoencoders can be very computationally expensive and typically require a large infrastructure for producing useful results from a huge corpus of content.

3. Novelty:
Recommendation engines are typically deployed where there is a huge influx of new content added all the time, where it can get almost impossible to manually search for engaging content, therefore the System ought to be responsive enough to manage user interests as well help discover new ones.

A great deal of research in matrix factorization methods [1] has been done, whereas research in deep neural networks for recommendation is significantly smaller. Apart from neural networks news recommendation [2], rating reviews [4], deep neural network based collaborative filtering [5], and music recommendation using deep neural networks [6], there seems to be a general lack of new and innovative research unlike other areas of deep learning.

II. SYSTEM DESIGN

The system is divided into two parts: one for the candidate generation and the other for scoring and ranking.This two-part modular approach allows us to make recommendations from a vast corpus of content while still ensuring that the small number of contents recommended is personalized and engaging. The overall architecture recommendation system is illustrated in Fig. 1.

The candidate generation takes available information and interests about groups of similar users as input and retrieves a small subset of generally relevant candidates from a large corpus. These candidates are intended to provide only broad personalization. Additionally, this modular design approach enables the blending of multiple candidate generation sources with existing filtering methods, such as those described in earlier work [7], to generate even more relevant results, which are then stored in a shared candidate pool.

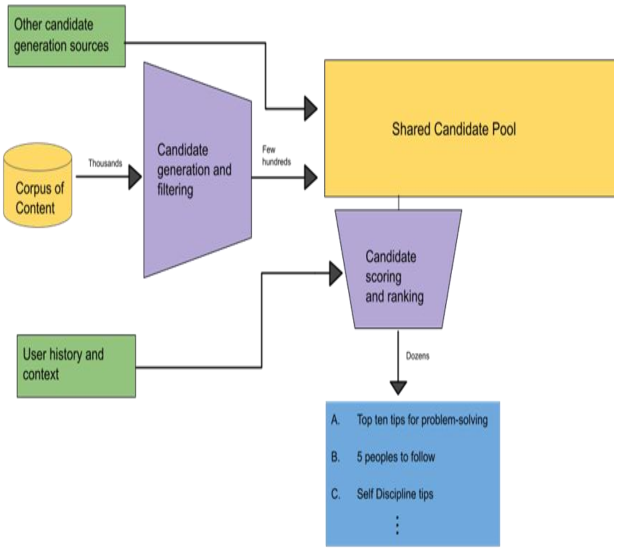

Fig 1. Recommendation system architecture describing a funnel-like filter and a shared pool where candidates are stored and are ranked specific to a user group.

the historical user behavior, due to sparseness and a range of unobservable external factors, is inherently tough to predict. we rarely acquire the ground truth of user satisfaction and model noisy implicit feedback instead.

Moreover, negative implicit feedback has natural scarcity, and metadata associated with the content is poorly structured without a well-defined ontology therefore the recommendation system needs to be robust to these particular characteristics of the data.

III. CANDIDATE GENERATION

P In this step, the system starts from a large corpus and generates a smaller set of candidates. The model must evaluate quickly given the large size of the corpus and generate relevant candidates. This system provides multiple candidate generators, each nominating a unique set of candidates, these different sets of the candidate are then stored as a cache in the pool which is to further improve efficiency, these candidates pools are updated as required.

1. Collaborative filtering with matrix factorization is employed as one of the source for candidate generation where the similarity between users is expressed in terms of coarse features like demographics, search queries, age, gender, while necessary these coarse features don't fully capture the similarity between users for that additional specific features are required such as the interaction between users and user-item interaction which is calculated based on direct and indirect user interaction with other users and items respectively. This interaction can be conceptualized as a graph where every node is a user and the weight of an edge connecting them is their degree of interaction, the higher the interaction more the weight, this works on the principle that similar people tend to interact with each other more.

2. Content-based filtering is also used as one of the sources for candidate generation. The main factor for determining the candidate will be keywords, which requires all the content to be first summarized into a helpful entity which will be generated by firstly converting any audio or video-based content into text, then that text is analyzed and inspected for any known entities which are then weighted between 0 and 1 (inclusively) according to their importance. These entities will be stored in the corpus itself with other metadata, this will also improve searchability by reducing its reliance on creator-provided metadata about content, as click-baiting and other such bad practices are on the rise which deceives users to get clicks, as well as help related users, discover new interests.

3. Additional Candidate Source:
Randomly sampled new item and popular item with varying degrees of similarity between items and user's metadata is inserted into the candidate generation pool to introduce nuance. Depending upon the circumstances even more candidate sources can be added.

By using multiple strategies for candidate generation, reliance on one source is reduced as this may introduce the bias towards a similar strategy e.g. A major drawback of a Content-based algorithm is that it is limited to recommending items that are too similar to each other. This inherent bias towards a similar strategy can be alleviated by using multiple methods for candidate generation. This also solves cold start by recommending popular regional content when there is no known user information, as the user interacts his/her preferences are known and recommendation gets better.

IV. SCORING AND RANKING

The ranking is also crucial while ensembling different candidate sources whose scores are not directly comparable.

Once all the candidates are stored in a shared candidate pool these are then fetched for scoring and ranking. The scoring is done based on historic user activity and user preferences. Whenever a user refreshes the page for a new recommendation it checks if the pool has been updated or not, if yes then, new candidates are fetched from the pool and scored. If not then, the model simply re-ranks previous results.

The primary role of ranking is to use personal preferences and events from the user's activity history to specialize and calibrate candidate predictions (in order of 10's) for the particular user. During ranking, it is possible to use more features because only a few hundred items are being scored rather than thousands.

Since the model only evaluates a small subset of items, the system uses a more precise model relying on additional queries. A fine representation is required to distinguish relative importance among candidates with high precision to present a few good recommendations in a list, ranking network accomplishes this task by taking into account any user-specified constraints such as explicit dislike or report/ban then these are removed from consideration; furthermore, all the remaining items in the pool are then allocated a score according to the specified objective function using a set of features describing the user and item. Finally, the few highest-scoring items are ranked presented to the user. We have consistently found that ranking content based on how much time a user has spent engaged with content [7, 8] instead of the item's click-through rate helps alleviate click-baiting or any other shady practices.

V. CONCLUSION

We have described our recommendation system architecture for personalizing content to users, split into two parts: candidate generation and ranking.

Initially, candidates are generated by using an ensemble of multiple candidate generation sources like collaborative filtering and content-based filtering systems. Using this two-part structure allows modularity, therefore other candidate generation sources can be easily added to best serve the users. Once the candidates are all generated they are stored in a shared candidate pool, which functions as a centralized cache for a specific set of similar people. This helps the system be more efficient and increases the serving speed.

The pool is shared among a specific group of users and it is used whenever a user from these groups requests for content is then retrieved and sent to the scoring and ranking model along with users' activity history and other user-specific data.

Finally, candidates are scored and ranked, then this final list of candidates is sent to the user. If the user refreshes his recommendation, rather than generating candidates from corpus the system will instead retrieve it from the cache, these candidates will be re-ranked with additional new content and presented to the user, which will significantly reduce the server processing.

Key takeaways and advantage of this approach can be summarized as:

- This content generation model can effectively incorporate many signals and model their interaction.
- The age of an item is considered just like any other feature while recommending which removes any bias towards the past.
- Ensemble multiple different candidate methods outperformed any single approach.
- By using a shared candidate pool the serving speed and efficiency of recommendation increased dramatically.
- The ranking is also crucial for assembling different candidate sources as it allows comparison of scores from all different candidate generation sources.
- Ranking by watch time yields much better results than ranking by click-through rate focuses on the

International Journal of Science, Engineering and Technology

An Open Access Journal

total time spent rather than how many clicks an item has attracted.

ACKNOWLEDGEMENTS

The authors would like to express their sincere gratitude to Prof. Vaishali Surjuse and all other teaching faculty of Karmaveer Dadasaheb Kannamwar Engineering College for their valuable guidance and support.

REFERENCES

[1] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. Advances in artificial intelligence, 2009:4, 2009.

[2] K. J. Oh, W. J. Lee, C. G. Lim, and H. J. Choi. Personalized news recommendation using classified keywords to capture user preference. In 16th International Conference on Advanced Communication Technology, pages 1283–1287, Feb 2014.

[3] D. Tang, B. Qin, T. Liu, and Y. Yang. User modeling with neural network for review rating prediction. In Proc. IJCAI, pages 1340–1346, 2015.

[4] H. Wang, N. Wang, and D.-Y. Yeung. Collaborative deep learning for recommender systems. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, pages 1235–1244, New York, NY, USA, 2015. ACM.

[5] A. van den Oord, S. Dieleman, and B. Schrauwen. Deep content-based music recommendation. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 26, pages 2643–2651. Curran Associates, Inc., 2013.

[6] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, and D. Sampath. The youtube video recommendation system. In Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10, pages 293–296, New York, NY, USA, 2010. ACM.

[7] E. Meyerson. Youtube now: Why we focus on watch time. http://youtubecreator.blogspo t.co m2012/08/youtube-now-why-we-focus-on-wat ch-time.html. Accessed: 2016-04-20.

[8] X. Yi, L. Hong, E. Zhong, N. N. Liu, and S. Rajan. Beyond clicks: Dwell time for personalization. In Proceedings of the 8th ACM Conference on Recommender Systems, RecSys '14, pages 113–120, New York, NY, USA, 2014. ACM.