RecDelta: An Interactive Dashboard for Cross-model Evaluation of Top-k Recommendation

Yi-Shyuan Chiang[*†]
Academia Sinica
Taipei, Taiwan

Yu-Ze Liu[*]
Academia Sinica
Taipei, Taiwan

Chen-Feng Tsai[‡]
Academia Sinica
Taipei, Taiwan

Jing-Kai Lou
KKStream Technologies
Taipei, Taiwan

Ming-Feng Tsai
National Chengchi University
Taipei, Taiwan

Chuan-Ju Wang
Academia Sinica
Taipei, Taiwan

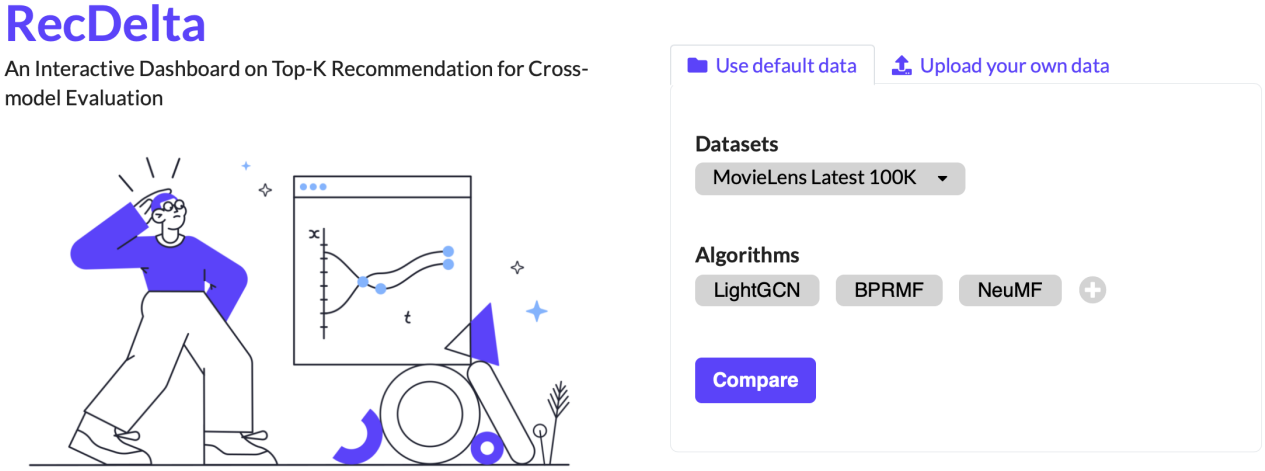

Figure 1: RecDelta homepage

ABSTRACT

In this demonstration, we present RecDelta, an interactive tool for the cross-model evaluation of top-k recommendation. RecDelta is a web-based information system where people visually compare the performance of various recommendation algorithms and their recommended items. In the proposed system, we visualize the distribution of the δ *scores* between algorithms—a distance metric measuring the intersection between recommendation lists. Such visualization allows for rapid identification of users for whom the items recommended by different algorithms diverge or vice versa; then, one can further select the desired user to present the relationship between recommended items and his/her historical behavior. RecDelta benefits both academics and practitioners by enhancing model explainability as they develop recommendation algorithms with their newly gained insights. Note that while the system is now online at https://cfda.csie.org/recdelta, we also provide a video recording at https://tinyurl.com/RecDelta to introduce the concept and the usage of our system.

CCS CONCEPTS

• **Information systems** → **Data analytics**.

KEYWORDS

Visualization, Recommender system, Explainability

ACM Reference Format:
Yi-Shyuan Chiang, Yu-Ze Liu, Chen-Feng Tsai, Jing-Kai Lou, Ming-Feng Tsai, and Chuan-Ju Wang. 2022. RecDelta: An Interactive Dashboard for Cross-model Evaluation of Top-k Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3477495.3531674

[*]Both authors contributed equally to this work.
[†]Currently a Ph.D student at University of Illinois Urbana-Champaign
[‡]Currently an undergraduate student at University of California, Berkeley

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531674

1 INTRODUCTION

Recommender systems are ever prevalent. With the help of machine learning, platforms such as YouTube, Amazon, or Netflix can provide personalized experiences [3, 10, 14]. For recommender systems to succeed, the algorithms used must make spot-on recommendations to ensure continued user interest. Typically, academics leverage aggregated evaluation metrics such as precision, recall, and normalized discounted cumulative gain (NDCG) to compare different recommendation algorithms. In addition to such evaluations on offline datasets, practitioners usually include eyeball evaluations before entering the next phase, e.g., A/B testing. Nevertheless, the unique nature of recommender systems—creating personalized recommendations for drastically different users—makes it difficult to evaluate different algorithms. When recommendations are closely tied to personal preferences, high aggregated evaluation scores do not necessarily reflect the actual quality of recommendations [9]. Moreover, when multiple algorithms share similar scores, it is difficult to determine which algorithms to use for recommendation in practice. In this case, practitioners must manually comb through the user's history and recommendations made by each algorithm to decide which model to adopt or how to combine the results from multiple algorithms. However, due to the nature of embedding-based or deep learning-based models, which generally are not explainable, analyzing the relationships among different algorithms becomes extremely challenging as the user and item embeddings from different algorithms are located in different embedding spaces and thus are not easily compared [5, 8, 11].

Therefore, evaluating recommender systems beyond typical evaluation approaches has long been a critical question for both researchers and machine learning practitioners. The literature contains unconventional evaluation methods proposed from various perspectives. Schröder et al. customize evaluation metrics for various fields, as not all projects share the same data usage profiles and goals [12], whereas Cremonesi et al. propose measuring users' perceived quality of recommendation and level of satisfaction instead of depending on accuracy metrics [1]. Fagin et al. quantify the extent of overlap between two lists to compare the intersection of the top-k lists among different algorithms [2]. To discover the relationship between items, Singh et al. incorporate cosine similarity into the development of KNN recommender systems [13]. However, most studies focus on quantitative evaluations only. To the best of our knowledge, there exists no visualization tool that compares multiple recommendation algorithms both quantitatively and qualitatively.

We present RecDelta, a prototyping system that visually compares the performance of various recommendation algorithms and their recommended items. Instead of comparing different algorithms in the unexplainable embedding spaces, we analyze the recommended lists generated from different algorithms. More precisely, RecDelta visualizes the distribution of δ *scores* between compared algorithms—a distance metric measuring the intersection between recommended lists, by which researchers can quickly identify users for whom the recommended items generated by different algorithms diverge or vice versa. After that, one can further select a user to inspect. The relationship among recommended items from different algorithms is illustrated by Venn diagrams, and the relationship between recommended items and historical behavior for the selected user is illustrated by heap maps. RecDelta's dynamic and interactive characteristics now make it possible to efficiently locate similarities and differences among recommendation lists and thus enhance model explainability. Furthermore, RecDelta provides valuable insights for constructing a better recommender system, including designing new algorithms, assembling results from multiple models, and tuning strategies for different sets of models.

2 SYSTEM DESCRIPTION

In this work, we present a tool featuring different model evaluation methods to simultaneously compare recommended items generated from multiple recommendation algorithms in visual analytic workflows. To showcase our idea, we conduct experiments and present visualizations on the MovieLens Latest 100K dataset [4].

2.1 Comparison of recommendation similarity using δ scores

In this study, we use the δ score defined in [2] to measure the difference between the two recommendation lists. Let ℓ_1 and ℓ_2 denote two top-k lists and $\ell_1(i)$ ($\ell_2(i)$) denote the truncation of ℓ_1 (ℓ_2) to the first i items. Given item sets $\mathcal{I}_{\ell_1(i)}$ and $\mathcal{I}_{\ell_2(i)}$ for $i = 1, \cdots, k$, each of which includes the items from $\ell_1(i)$ and $\ell_2(i)$, respectively, we have

$$\delta(\ell_i, \ell_j) = \frac{1}{k} \sum_{i=1}^{k} \left| \mathcal{I}_{\ell_1(i)} \Delta \mathcal{I}_{\ell_2(i)} \right| / (2i), \tag{1}$$

where the operator Δ is symmetric and $X \Delta Y := (X \setminus Y) \cup (Y \setminus X)$. For each user, δ describes the similarity between two top-k lists generated by two algorithms; the higher the δ score (ranging from 0 to 1), the lower the similarity.

In our system, we illustrate the distribution of δ of all users (see Figure 2), where each block represents a user and a bar is composed of multiple users. Note that as the δ score is derived from comparing lists, more than two algorithms (i.e., $n > 2$ algorithms) must be selected to generate δ score; for each user, there are C_2^n δ scores for all pairs of algorithms. Accordingly, the x-axis denotes the mean of the δ scores for each user, the color of which reflects the variance of the δ scores (note that blue denotes larger variance and light green smaller variance). Such a visualization provides a macroscopic perspective when inspecting differences among results from different recommendation algorithms. For example, one can easily observe that if the distribution plot is skewed to the left, the recommendations made by selected algorithms are generally dissimilar.

2.2 System interfaces and experience flows

RecDelta system has two primary interfaces: the main query portal and the comparison result page. Users specify the datasets and algorithms on the main query portal; the results are shown on the page they are later directed to.

2.2.1 Main query portal. In the main query portal, users either choose from our pretrained models or upload their own models with the dataset information. In this demonstration we provide five algorithms for users to choose from: LightGCN [5], BPRMF [11],

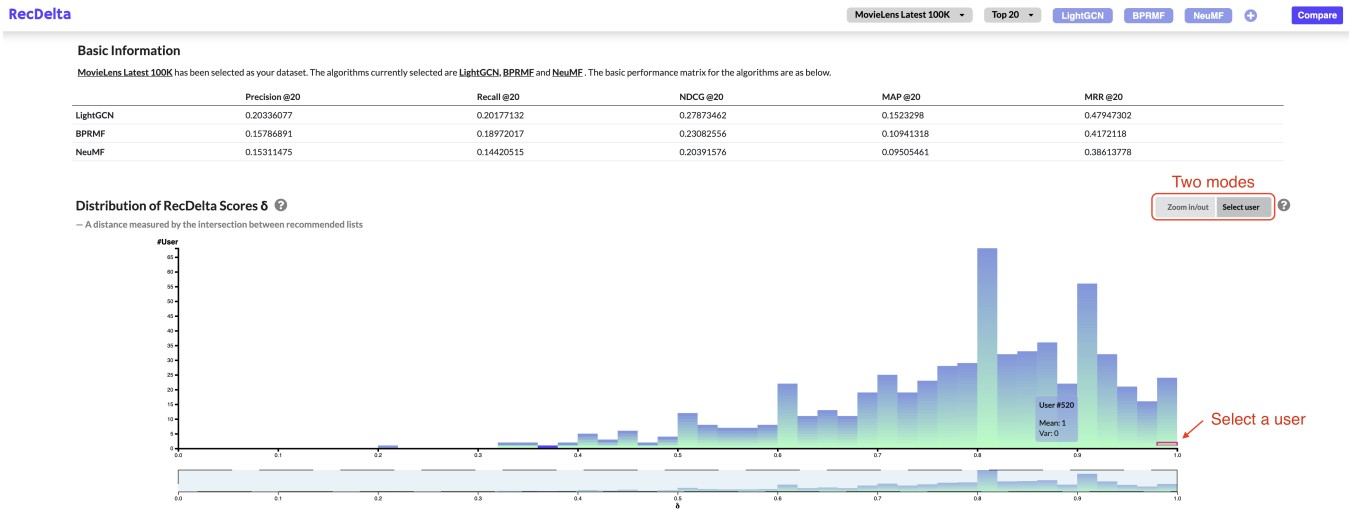

Figure 2: Distribution of δ scores

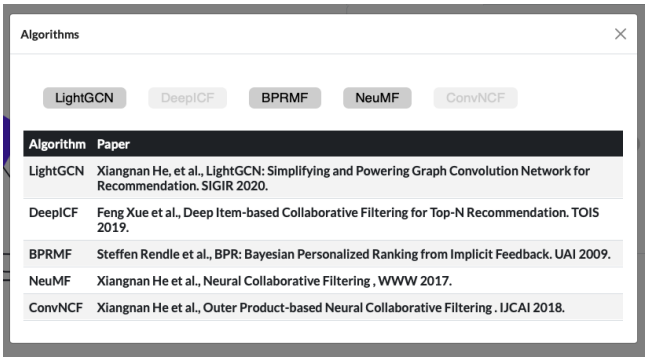

Figure 3: Pop-up window where users manage the algorithms they seek to inspect

NeuMF [7], DeepICF [15], and ConvICF [6]. Users manage the algorithms they seek to compare by clicking the "+" button (see the right panel in Fig. 1) and add more algorithms by selecting the algorithm names displayed on the pop-up window; algorithms can be removed by double-clicking the displayed names (see Fig. 3).

RecDelta also allows users to upload their own models with the dataset information. Users name and upload their own embeddings on the right tab for each compared algorithm. As the δ score describes the relationship between two or more algorithms, for each comparison, users must upload two or more sets of embeddings. Each set of embeddings contains item and user embeddings for a certain algorithm. Moreover, as we seek to two analyze the recommended items among different perspectives, users must also upload item metadata and user history. The data structure requirements are as follows. Item metadata should be in the `json` format and include "attr" and "data, " the former of which should have "name (string)," "numerical (list)," and "nominal(list)," and the latter of which should contain the "attr" for all items. The user history is also submitted as a `json` file, and includes the user-item interaction history, where

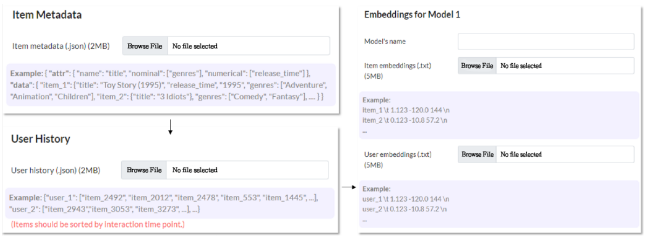

Figure 4: Model upload process. Users upload item metadata, user history, and embeddings.

items are sorted in chronological order according to the timestamp. Lastly, the user and item embeddings for each model should be in the `txt` format following the structure as the example, for instance, `user1 \t dim1 dim2 dim3 \n` (see, for example, Fig. 4).

2.2.2 Comparison result page. After the users have specified the algorithms they seek to compare on the main portal, they are directed to the comparison results page. On this page, RecDelta generates a comparison table consisting of the basic information for the algorithms chosen by the user, such as precision rates and recall rates. Below the chart is the distribution of the δ score. Users can hover on the "i" icon next to the title to assess the formula for the δ score. Users can switch between the *zoom* or *select* modes of the chart. Zoom mode allows users to zoom in on the chart to closely inspect data points. Once users have decided on the data point, they can select a data point—a user from the MovieLens Latest 100K [4]—in select mode.

Once the data point has been selected, RecDelta shows a Venn diagram that depicts the relations between each list of recommendations. Users gain an understanding of the relationship between each recommendation algorithm by observing how the recommendations overlap with each other in the diagram (see Fig. 5). The Venn diagram is interactive; users can click on any spot in the diagram to learn what the section covers. Items in the selected section are compiled into a list, where users can assess metadata by clicking

on an item of interest, and the metadata appears on the column on the right.

In addition to the Venn diagrams, we also provide heat maps to visualize the watch history of individual viewers compared to the algorithms' recommendations. The y-axis consists of the set of recommendations an algorithm makes, and the x-axis is the historical behavior of a data point. Each block represents the cosine similarity between a top-k recommended item and an item from the user's history. The darker the color of a block, the more similar the recommendation and the item from the viewer's watch history (see Fig. **??**). Displaying the watch history in a linear manner like this allows engineers to observe how each algorithm makes recommendations.

For example, each data point in the MovieLens Latest 100K dataset represents a single viewer; therefore, the heat map here represents the level of similarity between the recommendations and each viewer's movie watch history. If the user's history is categorized by genres in another dataset, each block represents the average of the cosine similarities between each top-k recommended item and all historical items belonging to a specific genre.

3 CASE STUDY

3.1 Venn diagrams: An intuitive way to interpret algorithm similarities

The δ scores reflect the levels of differences between the top-k recommendations made by different algorithms; a high δ indicates that the top-k recommendations of the two algorithms are dissimilar. When the algorithms are dissimilar, the overlapping areas in the Venn diagram should be little to none. Take user #294, for example: when comparing LightGCN, BPRMF, and NeuMF, user #294's δ score is on the higher end (0.85 out of 1). A score of 0.85 indicates that the recommendations made by the three algorithms are disjoint and share little to no similarity. This observation is illustrated by the non-overlapping Venn diagram shown in Fig. **??**. On the other hand, a low δ score for user #133, 0.25, indicates that the recommendation results mainly are very similar. From the Venn diagram in Fig. 5, it is clear that the algorithms share a large number of top-k items. When the recommendations are highly similar, the circles overlap with each other significantly.

3.2 Heat maps: A user-friendly way to visualize recommendations

The heat maps allow users to observe what aspects each algorithm focuses on. For user #294, the heatmaps show that LightGCN captures the user's long-term behaviors (see Fig. **??**), whereas BPRMF and NueMF put more weight on the user's recent behavior. It can also be helpful to be able to cross-reference the heat maps and see how different models complement each other. For example, in Fig. **??** it is clear what LightGCN and BPRMF emphasize and how they differ. Such visualization thus improves model explainability and provides valuable insights for recommender system improvements, for instance designing new algorithms, assembling results from multiple models, and tuning strategies for different sets of models.

4 CONCLUSION AND FUTURE WORK

In this demonstration, we present RecDelta, an interactive data visualization tool that incorporates performance metrics, Venn diagrams, and heat maps. Although for demonstration purposes, the current system provides only the MovieLens Latest 100K dataset along with five popular recommendation models for comparison, we include an option to upload custom models to make our solution more practical. In sum, RecDelta makes it easier for machine learning researchers and practitioners to investigate the results generated by various algorithms simultaneously from both macroscopic and microcosmic perspectives. For future work, we plan to include more embedding-based recommendation algorithms and datasets in the system. One other promising direction would be a command line tool for such cross-model comparisons.

REFERENCES

[1] Paolo Cremonesi, Franca Garzotto, Sara Negro, Alessandro Papadopoulos, and Roberto Turrin. 2011. Comparative Evaluation of Recommender System Quality. 1927–1932.

[2] Ronald Fagin, Ravi Kumar, and Dakshinamurthi Sivakumar. 2003. Comparing Top k Lists. *SIAM Journal on Discrete Mathematics* 17, 1 (2003), 134–160.

[3] Carlos A Gomez-Uribe and Neil Hunt. 2015. The Netflix Recommender System: Algorithms, Business Value, and Innovation. *ACM Transactions on Management Information Systems* 6, 4 (2015), 1–19.

[4] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems* 5, 4, Article 19 (2015), 19 pages.

[5] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 639–648.

[6] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. 2018. Outer Product-based Neural Collaborative Filtering. *arXiv preprint arXiv:1808.03912* (2018).

[7] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web*. 173–182.

[8] Tadiparthi VR Himabindu, Vineet Padmanabhan, and Arun K Pujari. 2018. Conformal Matrix Factorization Based Recommender System. *Information Sciences* 467 (2018), 685–707.

[9] Sean M McNee, John Riedl, and Joseph A Konstan. 2006. Being Accurate Is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems. In *Proceedings of the Conference for Human-Computer Interaction Extended Abstracts on Human Factors in Computing Systems*. 1097–1101.

[10] Bin Nie, Honggang Zhang, and Yong Liu. 2014. Social Interaction Based Video Recommendation: Recommending YouTube Videos to Facebook Users. In *Proceedings of the 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 97–102.

[11] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. 452–461.

[12] Gunnar Schröder, Maik Thiele, and Wolfgang Lehner. 2011. Setting Goals and Choosing Metrics for Recommender System Evaluations. In *Proceedings of the UCERSTI2 Workshop at the 5th ACM Conference on Recommender Systems*. 78–85.

[13] Ramni Harbir Singh, Sargam Maurya, Tanisha Tripathi, Tushar Narula, and Gaurav Srivastav. 2020. Movie Recommendation System Using Cosine Similarity and KNN. *International Journal of Engineering and Advanced Technology* 9, 5 (2020), 556–559.

[14] Brent Smith and Greg Linden. 2017. Two Decades of Recommender Systems at Amazon.com. *IEEE Internet Computing* 21, 3 (2017), 12–18.

[15] Feng Xue, Xiangnan He, Xiang Wang, Jiandong Xu, Kai Liu, and Richang Hong. 2019. Deep Item-based Collaborative Filtering for Top-*n* Recommendation. *ACM Transactions on Information Systems (TOIS)* 37, 3 (2019), 1–25.

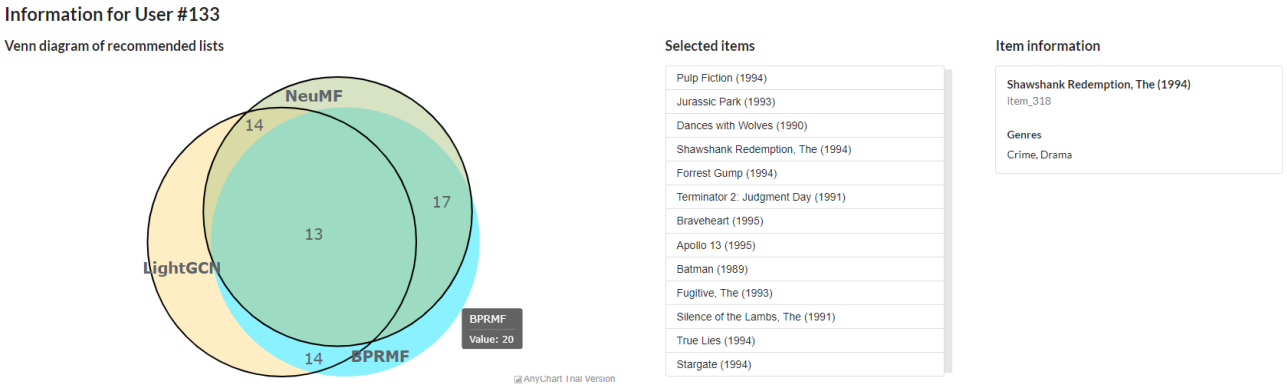

Figure 5: Information for a signal data point: A Venn diagram, a list of selected items, and the corresponding item metadata. By observing the overlap between the circles, users immediately perceive that the recommendation results from the three algorithms are similar.