RecForUS: A Recommender System for Uncertain Scores

Dvir Cohen
Technion – Israel Institute of Technology
dvir.cohen@campus.technion.ac.il

Liad Domb
Technion – Israel Institute of Technology
liad.domb@campus.technion.ac.il

Avigdor Gal
Technion – Israel Institute of Technology
avigal@technion.ac.il

Lior Ganon
Technion – Israel Institute of Technology
lior.ganon@campus.technion.ac.il

Eliezer Gavriel
Technion – Israel Institute of Technology
eliezerg@campus.technion.ac.il

Omri Lazover
Technion – Israel Institute of Technology
omri.lazover@campus.technion.ac.il

Coral Scharf
Technion – Israel Institute of Technology
coralscharf@campus.technion.ac.il

Bar Shterenberg
Technion – Israel Institute of Technology
sht.bar@campus.technion.ac.il

ABSTRACT

We present RecForUs, a recommender system designed to offer accurate music recommendations through a competition between participants and an algorithmic recommender. Our framework aims to demonstrate the intricate management of uncertain scores in a recommender system, catering to the specific objectives of users. The demonstration showcases our novel RankDist algorithm that efficiently computes rank probabilities for items with uncertain scores, enabling optimal selection of ranking semantics tailored to different user objectives without requiring exhaustive evaluation of all possible worlds. RecForUS is versatile, demonstrating the effectiveness of generating top-K query results in multiple scenarios.

PVLDB Reference Format:
Dvir Cohen, Liad Domb, Avigdor Gal, Lior Ganon, Eliezer Gavriel, Omri Lazover, Coral Scharf, and Bar Shterenberg. RecForUS: A Recommender System for Uncertain Scores. PVLDB, 18(12): 5267 - 5270, 2025.
doi:10.14778/3750601.3750648

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/liorgan93/demo_project.

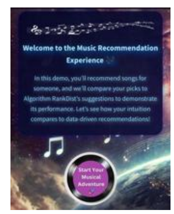

Figure 1: Opening Screen of the Demonstration.

1 INTRODUCTION

Recommender systems [4, 10, 14] provide an efficient vehicle to scalable data exploration. Top-K queries, prevalent in recommender systems, aimed at presenting users with a ranked list of items (a Top-K answer) in descending order of relevance [1]. Recommender systems rely on a scoring system, where users provide feedback, either implicit or explicit, on items with which they interact.

Uncertain scores. Scoring poses a challenge to recommender systems that need to correctly rank elements using scores that are often uncertain. Such uncertainty may be due to data unreliability, mainly attributed to user's feedback on items, which represents a subjective opinion that might be wrong or imprecise, and therefore, its trustworthiness is doubtful. Another source of uncertainty involves the difficulty in acquiring complete feedback data due to the inability of users to express their opinion on all items. Therefore, the knowledge about interactions of users with items is limited.

Motivation. There are few alternatives to cope with data uncertainty in recommender systems, among them ignoring uncertainty and removing uncertain values. The former may lead to unreliable results, while the latter may be costly and at times simply infeasible. Avoiding a proper analysis of uncertainty may cripple recommender systems in terms of quality performance leading to poor response to top-K queries. While several studies explore recommendations under uncertainty [3, 6, 8, 12], a gap remains in coherently associating between the optimization objective a top-K query should achieve and a proper ranking mechanism. In particular, a user may be interested in a top-K **set** of items or alternatively, value the relative ranking of items. As it so happens, different optimization goals possibly yield different top-K items with varying ordering.

Proposed solution. Our novel system, RecForUS demonstrates a recommender system for uncertain scores. It fills this gap using results from the work of Scharf *et al.* [9]. In particular, we demonstrate an approach that supports three real-world scenarios, each represents a user that seeks an answer to a top-K query with a different objective in mind. The three objectives are: 1) reaching perfect precision, as to avoid a high cost of false positives in sensitive job applications for interviews, 2) a set of relevant items, as

Proceedings of the VLDB Endowment, Vol. 18, No. 12 ISSN 2150-8097.
doi:10.14778/3750601.3750648

in choosing a playlist, and 3) an ordered list of items where positioning within the top-K matters, as in a search task. Each scenario is optimized by a different quality criterion that generates a high quality result and accordingly maximizes user satisfaction.

Demonstration. Our live RecForUS demonstration uses music playlists to engage participants in a fun competition against the RankDist algorithm [9] in offering an optimized response to a top-K query. Participants are presented with a sample set of songs. Based on their response, participants are positioned in a vector space (illustrated in Figure 2). Participants are then matched with an avatar with similar taste in music (measured using the vector space) and are asked to create a list of recommendations based on a specific optimization objective. The demonstration is concluded by comparing the participant and algorithmic recommendations. RecForUS integrates participant feedback with algorithmic analysis, providing participants with valuable insights into the algorithm's task and performance. This interactive session allows participants to experience first-hand the intricacy of recommendations under uncertainty, as detailed in Section 2. Our demonstration uniquely illustrates the translation of theoretical results into practice, providing an interactive platform where participants experience the impact of ranking semantics on recommendation quality while competing against a provably optimal algorithm.

Related work. The algorithmic solution, proposed by Scharf *et al.* [9] makes use of semantics for top-K querying [5, 13], matching them with appropriate objectives, such as precision-at-K and DCG-at-K. Numerous studies address uncertainty in recommender systems by investigating techniques to estimate and evaluate user-item score probabilities, as well as leveraging both scores and uncertainties for top-K recommendations [7, 8]. Rank-based semantics, which are uncommon in the literature of uncertain scores in recommender systems [3] were shown to offer significant value in top-K queries [9]. In this demonstration we show how to match a participant with an avatar with similar taste in music using vector representation, and offer a benchmark for comparing algorithmic results with participant recommendations.

2 APPROACH

We first present our approach in classifying a demonstration participant and selecting an avatar to match her music taste (Section 2.1). Then, for the purposes of this demonstration we summarize the approach of top-K recommendations over uncertain scores, as presented by Scharf *et al.* [9] (Section 2.2).

2.1 Modeling Participant Taste in Music

We model participants' musical preferences using a vector space that represents user behavior and taste based on their interaction history. This vector space allows identifying users with similar preferences and match participants with pre-defined musical avatars.

The modeling process begins by aggregating listening data from the user-song interaction logs.[1] Each user is represented by a numerical feature vector derived from their engagement with different songs. We apply standard preprocessing steps, including data cleaning, normalization, and filtering users based on a minimum threshold of interaction data.

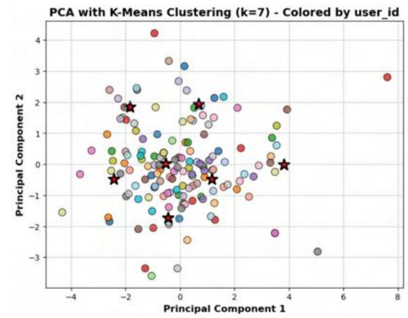

Figure 2: PCA With K-Means Clustering Centroids.

We constructed the vector space by representing each song using its audio features - such as genre, acousticness, and danceability. Playlists were then aggregated by averaging the audio feature vectors of all songs they contain, creating a compact representation of their overall musical profile.

To transform high-dimensional user data into a tractable representation, we use Principal Component Analysis (PCA) for dimensionality reduction. PCA helps project the original data into a lower-dimensional space that captures the most significant variance in user preferences. This enables both visual interpretability and effective clustering. Figure 2 provides an illustration of representative data points from the listening dataset.

Dimensionality reduction is followed by clustering using K-Means to segment users into distinct clusters, each representing a different musical profile. Each cluster centroid corresponds to a canonical vector that defines the taste of a virtual avatar. We use red ⋆ symbol in Figure 2 to represent cluster centroids.

When a new participant interacts with the system, their responses are encoded into the same PCA-transformed space, and they are assigned to the nearest avatar cluster based on Euclidean distance. This methodology ensures that participants are matched with avatars whose preferences reflect similar musical inclinations, creating a consistent and personalized recommendation experience.

The implementation of PCA and K-Means clustering is available in our public GitHub repository.[2]

2.2 Recommendations with Uncertain Scores

Top-K queries in recommender systems rely on an input dataset that contains information about users, items, and their interactions. Figure 3 [9] provides an overview of the key processing steps of top-K queries with uncertain scores in recommender systems.

Let U_I and U_U denote the universe of M items and Q users, where $p_i \in U_I$ and $u_j \in U_U$ represent a single item and a single user, respectively. User's scores of items are recorded in a database D. Each recorded interaction is stored as a tuple consisting of an item p_i, a user u_j, a score s selected from a predefined set of possible values, *Scores*, and potentially additional information.

User feedback is inherently subjective and influenced by cognitive biases and noise, leading to uncertain scores. To account for

[1] CSV files were extracted from the Kaggle notebook: https://www.kaggle.com/code/ahmedaliraja/music-recommendation-system-collabrating-filtering/notebook

[2] https://github.com/liorgan93/demo_project

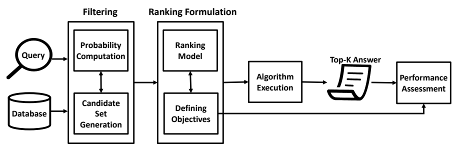

Figure 3: Top-K processing pipeline for generating high-quality recommendation from uncertain scores [9].

uncertainty, the attribute-level uncertainty model from probabilistic databases [11] is adopted. Rather than assigning a fixed score, the probability computation component (see Figure 3) is in charge of generating a probability distribution $S(p_i, u_j)$ for each user-item pair using a machine learning approach. The candidate set generation step narrows the search space by applying filtering techniques that select a smaller subset of items most likely to be relevant for users. The output of the filtering step is a set of items U to be ranked for each user u_j, along with score distribution $S(\sigma)\ \forall \sigma \in U$.

With deterministic scores, we simply sort items in a descending order. However, uncertain scores lead to a wide range of alternative semantics. The ranking model may compute a single representative score for each candidate item (*e.g.*, expected score), from which a ranking is derived. Alternatively, probability distributions of item scores may be utilized, allowing the use of diverse score-based and rank-based top-K semantics. Score-based semantics derive rankings directly from item score distributions, while rank-based semantics consider the probabilities of items appearing at specific ranks due to score uncertainty. Two example ranking semantics (taken from the probabilistic ranking literature) are the Global top-K approach, which selects the K items most likely to be ranked within the top-K positions [15], and the Expected score method, which selects the K items with the highest expected scores [2].

Given multiple ranking semantics, a crucial question is how to select the most suitable approach for a given recommendation task. The paper suggests that ranking semantics should optimize a selected objectives, thus ensuring high-quality results and maximizing user satisfaction. Three quality measures guide the semantics selection, namely *P@K*, *P@K*(K) and *DCG@K*, each represents a different typical real-world scenario. P@K is a rank-based measure focused on assessing the system's success in retrieving relevant items. It serves as a quality measure when recommending a **set** of relevant items, such as music playlists. P@K(K) quantifies the probability of achieving a perfect P@K, indicating scenarios where perfect precision is critical, such as avoiding false positives in sensitive job application processes. Finally, DCG@K measures the quality of rankings when both the relevance and the ordering of recommended items matter, as is common in search engine results.

For each of these metrics, Scharf *et al.* [9] characterizes selection criteria that provably optimize their expectations. Specifically, it demonstrates that the *Global top-K* semantics maximizes the expected *P@K* by selecting the K items most likely to appear in the top-K positions. The *U-TopK* semantics is shown to optimally maximize *P@K*(K) by selecting K items that have the highest probability of exactly matching the top-ranked items across all possible ranking outcomes. Lastly, the *Expected Score* semantics optimizes *DCG@K*, by selecting the K items with the highest expected scores.

Scanning all possible worlds is obviously infeasible. To efficiently support the computation of multiple rank-based ranking semantics, the paper introduces the *RankDist* algorithm, designed for computing the probability distribution of item rankings. The algorithm receives a candidate set of items along with their score distributions as input and calculates, for each item, the probability of being ranked at every possible position within the top-K rankings.

3 SYSTEM AND DEMO OVERVIEW

We demonstrate RecForUS, an entertaining tool for exploring recommendations using uncertain scores through a music recommendation scenario (see Figure 4).

3.1 Demonstration Flow

In this section, we delve into the application itself, presenting all of its core components. Each part is described both in terms of its functionality and its intended purpose within the user experience.

3.1.1 ***Introduction and Participant Profiling.*** The first stage of RecForUS introduces the participant to the recommendation experience and begins the process of profiling their musical preferences. Users are presented with a selection of songs to which they can listen and rate by indicating whether they like or dislike each track.

This interaction serves two goals. First, the participant's taste is modelled as a vector in a vector space to be matched with an avatar with similar taste in music, for generating personalized recommendations. Second, this stage enhances interactivity and engagement. By allowing participants to immediately immerse themselves in the task through music playback and rating, RecForUS fosters a clear and intuitive understanding of its goal and flow.

3.1.2 ***Getting to Know the Avatar.*** At this stage, the participant is introduced to an avatar selected based on their musical preferences identified in Section 3.1.1. The participant now takes on the role of a recommender system for this avatar.

To assess compatibility, the participant indicates her familiarity with songs that the avatar enjoys. If the participant is unfamiliar with more than 50% of the songs, the system assigns a different avatar, which better aligns with the participant's musical taste.

This step serves three key purposes. First, it verifies that the assigned avatar shares a sufficiently similar musical style with the participant, ensuring that the recommendations will be relevant. Second, it helps the participant to get to know the avatar's preferences, simulating the essential learning process that any recommender system must perform. Third, it sustains user engagement by continuing the interactive experience through active listening and decision-making, enhancing the participant's sense of immersion and understanding of the avatar's musical profile.

3.1.3 ***Recommender System Choice & Participant's Performance Comparison.*** After confirming that the participant is familiar with the avatar's musical preferences (as established in Section 3.1.2), they now present the avatar with a recommendation.

The participant selects one of three recommendation objectives (see Section 2.2), namely Perfect Precision, Relevant Set, or Ordered List. Then, the participant is presented with a list of songs and can listen to them before making a selection. Based on the selected objective, the participant recommends songs to the avatar.

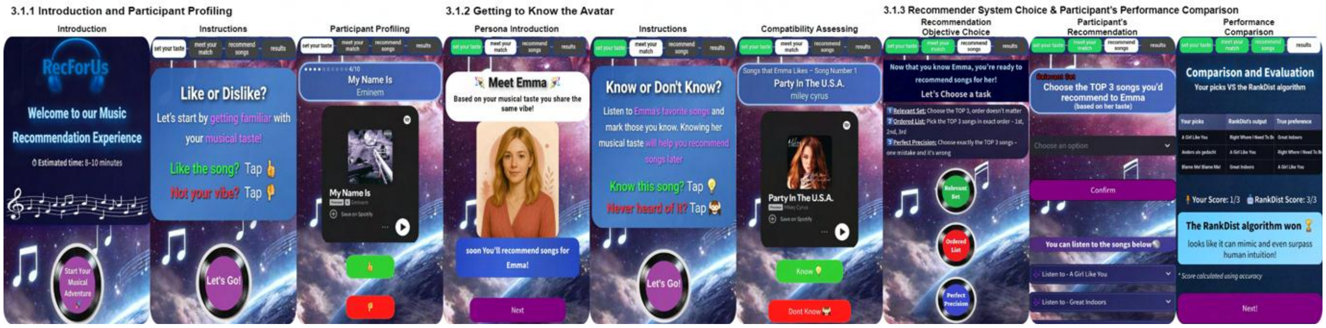

Figure 4: Demonstration flow

The participant's recommendations are then directly compared with those produced by the algorithm. The system provides clear feedback on the degree of similarity or difference between the human-generated and algorithmic suggestions.

This is the core part of the demonstration. After all prior preparation, the participant now acts as a functioning recommender system. By comparing the participant's performance with that of the algorithm, we gain valuable insights into the system's accuracy and how closely its recommendations align with human decision-making.

3.2 Technical Details

Participants will access the RecForUS demonstration by scanning a QR code, launching the demo on their personal mobile device. The system is fully compatible with both iOS and Android platforms. Throughout the experience, participants will listen to music tracks using personal headphones connected to their mobile phones.

We use Streamlit, a Python framework for developing interactive web applications. It was complemented by HTML and CSS to enhance layout and visual design. The system also uses core Python libraries such as pandas for data handling, and MP3 files were integrated to enable music playback directly within the application.

4 CONCLUSION

This demonstration presents RecForUS, a novel recommender system for uncertain scores that bridges the gap between optimization objectives and ranking semantics in top-K queries. By engaging participants in an interactive competition against the RankDist algorithm through music recommendations, we showcase how different user objectives, namely perfect precision, relevant set selection, and ordered ranking, can be effectively optimized using appropriate semantics. The demonstration not only illustrates the theoretical underpinnings of handling uncertain scores in recommender systems but also provides valuable insights into how algorithmic recommendations align with human intuition. Through our carefully designed user journey of profiling, avatar matching, and comparative recommendation, RecForUS offers conference attendees a hands-on experience with the challenges and solutions in managing uncertain scores while highlighting the importance of selecting appropriate ranking semantics based on specific user needs. Moreover, by demonstrating how different optimization objectives directly influence recommendation outcomes, RecForUS empowers users to make informed choices about their recommender systems, leading to more personalized experiences that align with their specific needs, whether they seek perfect precision for critical applications, diverse relevant content for entertainment, or carefully ordered results for efficient information retrieval. The system's modular design also makes it adaptable to other domains beyond music recommendations, opening avenues for future research in uncertainty-aware recommender systems.

REFERENCES

[1] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender systems survey. *Knowledge-based systems* 46 (2013), 109–132.

[2] Graham Cormode, Feifei Li, and Ke Yi. 2009. Semantics of ranking queries for probabilistic data and expected ranks. In *2009 IEEE 25th International Conference on Data Engineering*. IEEE, 305–316.

[3] Victor Coscrato and Derek Bridge. 2023. Estimating and Evaluating the Uncertainty of Rating Predictions and Top-n Recommendations in Recommender Systems. *ACM Transactions on Recommender Systems* (2023).

[4] 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.

[5] Ihab F Ilyas and Mohamed A Soliman. 2011. Probabilistic ranking techniques in relational databases. *Synthesis Lectures on Data Management* 3, 1 (2011), 1–71.

[6] Norman Knyazev and Harrie Oosterhuis. 2023. A Lightweight Method for Modeling Confidence in Recommendations with Learned Beta Distributions. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 306–317.

[7] Yehuda Koren and Joe Sill. 2011. Ordrec: an ordinal model for predicting personalized item rating distributions. In *Proceedings of the fifth ACM conference on Recommender systems*. 117–124.

[8] Fernando Ortega, Raúl Lara-Cabrera, Ángel González-Prieto, and Jesús Bobadilla. 2021. Providing reliability in recommender systems through Bernoulli matrix factorization. *Information sciences* 553 (2021), 110–128.

[9] Coral Scharf, Carmel Domshlak, Avigdor Gal, and Haggai Roitman. 2025. A Rank-Based Approach to Recommender System's Top-K Queries with Uncertain Scores. *Proc. ACM Manag. Data* 3, 1 (2025), 5:1–5:26. https://doi.org/10.1145/3709655

[10] Harald Steck. 2015. Gaussian ranking by matrix factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*. 115–122.

[11] Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch. 2011. Probabilistic databases. *Synthesis lectures on data management* 3, 2 (2011), 1–180.

[12] Chao Wang, Qi Liu, Runze Wu, Enhong Chen, Chuanren Liu, Xunpeng Huang, and Zhenya Huang. 2018. Confidence-aware matrix factorization for recommender systems. In *Proceedings of the AAAI Conference on artificial intelligence*, Vol. 32.

[13] Yijie Wang, Xiaoyong Li, Xiaoling Li, and Yuan Wang. 2013. A survey of queries over uncertain data. *Knowledge and information systems* 37, 3 (2013), 485–530.

[14] Markus Weimer, Alexandros Karatzoglou, Quoc V Le, and Alex J Smola. 2008. Cofi rank-maximum margin matrix factorization for collaborative ranking. In *Advances in neural information processing systems*. 1593–1600.

[15] Xi Zhang and Jan Chomicki. 2009. Semantics and evaluation of top-k queries in probabilistic databases. *Distributed and parallel databases* 26, 1 (2009), 67–126.