ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/262410817

Learning to rank for recommender systems

Conference Paper · October 2013

DOI: 10.1145/2507157.2508063

CITATIONS
24

READS
1,821

3 authors, including:

Alexandros Karatzoglou
Telefónica I+D
79 PUBLICATIONS **2,260** CITATIONS

SEE PROFILE

Yue Shi
The Hong Kong University of Science and Tech…
36 PUBLICATIONS **806** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project

Deep learning for recommender systems View project

All content following this page was uploaded by Alexandros Karatzoglou on 30 September 2015.

The user has requested enhancement of the downloaded file.

Learning to Rank for Recommender Systems

Alexandros Karatzoglou[1], Linas Baltrunas[1], Yue Shi[2]
[1]Telefonica Research, Spain; [2]Delft University of Technology, Netherlands;
[1]{alexk, linas}@tid.es, [2]y.shi@tudelft.nl

ABSTRACT

Recommender system aim at providing a personalized list of items ranked according to the preferences of the user, as such ranking methods are at the core of many recommendation algorithms. The topic of this tutorial focuses on the cutting-edge algorithmic development in the area of recommender systems. This tutorial will provide an in depth picture of the progress of ranking models in the field, summarizing the strengths and weaknesses of existing methods, and discussing open issues that could be promising for future research in the community. A qualitative and quantitative comparison between different models will be provided while we will also highlight recent developments in the areas of Reinforcement Learning.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information Filtering, Retrieval Models*

Keywords

Collaborative filtering, learning to rank, ranking, recommender systems

1. INTRODUCTION

Recommender systems aim to provide users with personalized items, which are typically ranked in a descending order of predicted relevance [1]. Learning the personalized recommendation list can be cast as a ranking problem. Naturally this cutting-edge research topic, *Learning to Rank* (LtR) [4], has already attracted a lot of attention in the Information Retrieval and Machine Learning communities. Recent contributions to collaborative filtering (CF) have exploited LtR techniques for improving the ranking of the top-N recommendations. In this tutorial, we present the key ideas of different categories of learning to rank approaches, and demonstrate examples that extend these ideas to specific CF methods. We also discuss a few open issues that remain challenging for future research in this direction.

2. OVERVIEW

2.1 Background

The tutorial briefly introduces the background of recommender systems and CF techniques. In particular, conventional CF methods target the rating prediction problem, such as the problem defined in Netflix Prize contest [1]. However, we emphasize that the more important objective in recommender systems is *ranking*, or the *top-N recommendation.* We also review some conventional ranking methods for recommendation, such as item-based CF [8].

2.2 LtR for Recommender Systems

We introduce the concept of LtR in the area of information retrieval, and explain its usefulness for recommender systems based on the analogy between query-document search and user-item recommendation. Then, we review three types of LtR techniques, i.e., point-wise, pair-wise and list-wise LtR.

CF methods that learn a ranking model based on the preference scores of individual items can are considered point-wise ranking methods [3]. With pair-wise LtR, CF methods can be developed to take into account the preferences of each user to a pair of items. A typical example of CF with paire-wise LtR is Bayesian personalized ranking [7]. CF methods based on list-wise LtR model the list-wise preferences of each user to a list of items (usually the rated items). An important branch of CF methods in this category are designed to directly optimize the evaluation metrics, such as mean average precision (MAP), Mean Reciprocal Rank (MRR) and Normalized Discounted Cumulative Gain (NDCG), which are usually list-wise ranking measures. Typical examples of those methods are CofiRank [12], CLiMF [10] and TFMAP [9].

2.3 Implicit and Explicit Feedback

The core idea behind CF is that users whose past interests were similar will also share common interests in future. The user's interest is inferred by the user interaction patterns with the items either explicitly or implicitly. In explicit feedback, users are asked/allowed to explicitly rate the items that have been purchased/consumed, using a pre-defined Likert scale (graded relevance), e.g., 1-5 stars in

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s). Copyright is held by the author/owner(s).
RecSys'13, October 12–16, 2013, Hong Kong, China.
ACM 978-1-4503-2409-0/13/10.

[1]http://www.netflixprize.com/

Netflix movie recommendation site. Higher grade indicates higher preference/relevance of the item. In implicit feedback [2], users interact with items by downloading, purchasing, and their preferences are deduced from the interaction patterns. In this paper we propose a new model for the case of explicit/graded relevance data.

Various models for implicit feedback data use learning to rank [4] techniques to optimize binary relevance data ranking metrics. For example, several CF models [7, 9, 10] compute near optimal ranked lists with respect to the Area Under the Curve (AUC), Average Precision (AP) [5] and Reciprocal Rank [11] metrics. However, metrics that are defined to handle binary relevance data are not directly suitable for graded relevance data. Binary metrics, and CF methods that optimize for these metrics can be used on graded relevance data if it is converted to binary relevance by e.g., imposing a threshold (e.g., setting rating 4 as the threshold for the 1-5 scale so that items rated 4 and 5 are treated as relevant). This process has two major drawbacks: 1) we *lose grading information* within the rated items, e.g., items rated with a 5 are more relevant then items rated with a 4. This information is crucial in building precise models. 2) the choice of the *thresholding relevance is arbitrary* and will have an impact on the performance of different recommendation approaches.

2.4 Interactive Recommendation

In many recommendation domains the content is often dynamic and/or short lived such as video, news recommendation and computational advertising. In these domains it is often difficult to collect enough information in the form of clicks or ratings to recommend an item accurately using standart CF methods. Exploration/exploitation methods from reinforcement learning such as Multi-armed Bandits are particularly well suited for these domains. LtR techniques can be also used for Multi-armed Bandits [6].

2.5 Open Issues

While there has been significant progress in the development of novel ranking methods one significant issue in CF methods and recommender systems in general is modeling the dependencies of the items in a recommendation list. Modeling the dependencies within a recommendation list can potentially produce more accurate ranking, increased diversity and novelty. Other open issues despite significant progress include the efficient modeling of context (e.g. location) and item content information in ranking models.

3. OBJECTIVES

The topic of this tutorial focuses on the cutting-edge algorithmic development in the area of recommender systems. This tutorial would bring a big picture of the progress of ranking models in the field, summarizing the strengths and weaknesses of existing methods, and discussing open issues that would be promising for the future research in the community. The tutorial is intended for researchers and practitioners in the area of recommender systems, especially those who are interested in recommendation algorithms.

4. ACKNOWLEDGEMENTS

This work is funded as part of a Marie Curie Intra European Fellowship for Career Development (IEF) award (CARS, PIEF-GA-2010-273739) held by Alexandros Karatzoglou.

5. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Kowledge and Data Engineering*, 17(6):734–749, 2005.

[2] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. ICDM '08, pages 263–272. IEEE Computer Society, 2008.

[3] Y. Koren and J. Sill. Ordrec: an ordinal model for predicting personalized item rating distributions. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '11, pages 117–124, New York, NY, USA, 2011. ACM.

[4] T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[5] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval.* Cambridge Univ. Press, Cambridge [u.a.], 1. publ. edition, 2008.

[6] F. Radlinski, R. Kleinberg, and T. Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 784–791, New York, NY, USA, 2008. ACM.

[7] S. Rendle, C. Freudenthaler, Z. Gantner, and S.-T. Lars. Bpr: Bayesian personalized ranking from implicit feedback. UAI '09, pages 452–461. AUAI Press, 2009.

[8] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. WWW '01, pages 285–295. ACM, 2001.

[9] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, and N. Oliver. TFMAP: optimizing map for top-n context-aware recommendation. SIGIR '12, pages 155–164. ACM, 2012.

[10] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver, and A. Hanjalic. CLiMF: learning to maximize reciprocal rank with collaborative less-is-more filtering. RecSys '12, pages 139–146. ACM, 2012.

[11] E. M. Voorhees. The trec-8 question answering track report. In *TREC-8*, 1999.

[12] M. Weimer, A. Karatzoglou, Q. Le, and A. Smola. Cofirank - maximum margin matrix factorization for collaborative ranking. NIPS'07, pages 1593–1600, 2007.

View publication stats