Moving beyond Linearity and Independence in Top-N Recommender Systems

Evangelia Christakopoulou
Email: evangel@cs.umn.edu

ABSTRACT

This paper suggests a number of research directions in which the recommender systems can improve their quality, by moving beyond the assumptions of linearity and independence that are traditionally made. These assumptions, while producing effective and meaningful results, can be suboptimal, as in lots of cases they do not represent the real datasets. In this paper, we discuss three different ways to address some of the previous constraints. More specifically, we focus on the development of methods capturing higher-order relations between the items, cross-feature interactions and intra-set dependencies which can potentially lead to a considerable enhancement of the recommendation accuracy.

1. INTRODUCTION

During the last years, the area of top-N recommendation has received great attention and become widespread [3]. The methods developed [7, 8, 10, 11, 13, 14] provide better recommendations as the years go by. However, there are still some simplifications assumed that might not always represent reality. When this is the case, the methods developed clearly suffer from the ignorance of this valuable information, that is anyway present in the dataset.

The first assumption that is commonly made is the one of independence. In a k-nearest neigbors system [8], it is assumed that every item that the user has purchased independently contributes to the rating score that the user would give to another target item. This assumption holds in a lot of commercial modern systems, like the ones suggesting a set of songs to a user, based on his listening history [5].

Moving beyond independence, the vast majority of recommender systems developed are still simple; they mostly create linear models. An example of this is a modern top-N recommendation method [11] that performs better than other algorithms and it is based on a linear regression model. In the same way, when side information is taken into account, it still exploits linear relations between the features [4, 12, 17].

RecSys'14 October 06 - 10 2014, Foster City, Silicon Valley, CA, USA

ACM 978-1-4503-2668-1/14/10 ...$15.00.
http://dx.doi.org/10.1145/2645710.2653361

We think that these assumptions while providing us with easy-to-interpret and simple models, they set a limit to the capabilities of recommender systems. By allowing recommender systems to develop more complex methods that will better capture real-life data, we may be able to get more effective and powerful recommendations. It should be noted that such approaches have already been introduced [15, 16] for rating prediction methods. However, this is not the case for top-N recommendation methods. Also, it has been shown [7] that top-N recommendation methods and rating prediction methods exhibit different characteristics, so a simple extension would not suffice. Towards this direction, in this work, we will investigate the following research questions:

- Do higher-order relations exist in the rating information in datasets, beyond pairwise associations? If yes, which is the best way to incorporate them?
- Which is the best way to incorporate side information, apart from a linear model? Is cross-feature interaction present? How can we best model it?
- In top-N recommendation lists, it is assumed that every item in the list is the best possible, independent of the others, without however guaranteeing the best possible list. How can we optimize this, so that the overall list is the best?

The way we have addressed or plan to address the above research questions is analyzed in the following sections. Initial results [6] show that more complex interactions exist in the datasets and when taken into account, they can improve the recommendation quality.

2. HIGHER-ORDER RELATIONS

Current top-N recommendation methods compute the recommendations by taking into account only relations between pairs of items. However, when higher-order relations between items exist, the above approach is suboptimal. In [6], we propose a way (HOSLIM) to incorporate higher-order relations in recommender systems.

More specifically, we first verify the existence of higher-order relations in real-world datasets, by measuring how prevalent are the itemsets with strong association beyond the items that comprise it (beyond pairwise associations). We use different metrics to measure the strength of the associations

and different ways to measure how well the itemsets with strong association cover the datasets. From our analysis, we conclude that the presence of higher-order relations in different real-world datasets is clear and it cannot be ignored.

We then specify the itemsets, i.e. the sets of items that are co-purchased by at least σ users in the dataset. We compute the likelihood that a user will buy a specific item as a sparse aggregation of both the items and the itemsets purchased by the user. Our method builds upon the SLIM method [11] which estimates the rating that a user would give to an item as a sparse aggregation of the items rated by the user:

$$\tilde{R} = RS, \tag{1}$$

where R is the user×items rating matrix and S is the sparse aggregation coefficient matrix learned, that represents the similarities between the items.If R' denotes the users×itemsets matrix containing the information whether a user purchased all the items constituting an itemset, then the model proposed by our apporach is:

$$\tilde{R} = RS + R'S'. \tag{2}$$

The sparse matrix representing the similarities between items and itemsets (S') as well as S are learned, by solving an l_1 and l_2 regularized optimization problem. These matrices are used during the prediction time, in order to recommend the best top-N items to a user, by taking into account the higher-order information captured by the itemsets-items interaction.

After doing a full parameter study, exploring different levels of sparsity as well as different values of support threshold, we saw that the proposed method outperforms a variety of other top-N recommendation methods: the traditional and popular k-nearest neighbors [8], SLIM which is a modern top-N recommendation method outperforming the rest of the existing top-N methods [11] and HOKNN, which is the method that was the first to incorporate itemsets into the recommendation [8]. On top of that, we showed that there is a strong positive correlation between the existence of higher-order relations in a dataset and the performance of the proposed method, which demonstrates that our method can indeed capture effectively the existence of higher-order information and use it to improve the recommendation quality. In addition, we suggested a constrained version of our method, which controls the complexity and provides more efficient recommendations. By introducing this constraint, the recommendation performance is either not affected at all in some cases, or slightly decreased but still higher than the performance of the competing methods. In this way, our method can be easily used in order to provide good-quality recommendations fast.

The main purpose of this work was to answer the research question whether higher-order information exists in real-world datasets and whether incorporating it could help the recommendation quality. The take-away message was that by coupling the incorporation of higher-order associations (when they do exist) with a state-of-the-art top-N method, the quality of the recommendations made is improved beyond the current best results.

3. CROSS-FEATURE INTERACTIONS

Modern recommender systems in general also have side information, except from the traditional rating data. Side information could be user-based, like user demographics (e.g. age, gender, ethnicity, country of residence, user interests), or item-based (e.g.product characteristics), or both. A lot of recommender systems also have reviews , which are associated with the ratings themselves. By aggregating all the reviews written about a specific item, we have item side information and similarly by aggregating all the reviews written by a specific user, we have user side information. Different recommendation methods [4,12,17,19] take into account side information, as it enriches the ratings with additional features which if used properly can help improve the recommendation quality.

The existing approaches for incorporating side information exploit linear relations between the features. This means that the rating a user would give to an item is predicted as a linear combination of the different features of the user/item side information. However, these methods do not incorporate the interaction between the features. For example, let's suppose that a user is interested in articles related to 'Recommender Systems'. This feature is indicative of the fact that he will like articles related to 'Collaborative Filtering' or 'Matrix factorization'. Such cross-feature interactions are often not explicitly modeled by the methods.

In the work we are planning to do, we would like to better capture such cross-feature interactions, in the form of a bilinear model. More specifically, if I (items×item-features) is the matrix capturing item-side information, we will learn a matrix F (item-features×item-features) that will capture the cross-feature interaction. Specifically, the model used to estimate the ratings that users will give to items will be:

$$\tilde{R} = RIFI^T. \tag{3}$$

It is worth noting that RI basically represents the user side information, as for every user, his side information is the aggregation of the item side information of the items he rated.

In the same way, we can also incorporate user side information, too. If U (users×user-features) is the matrix capturing user-side information, then we can learn a matrix W (user-features×user-features) that will also capture the cross-feature interaction, for the user side now. Then, the model will be:

$$\tilde{R} = UWU^T R. \tag{4}$$

We can either use Equations 3 and 4 individually, or we can combine them (depending on how sparse our datasets are) to get

$$\tilde{R} = \alpha RIFI^T + (1-\alpha)UWU^T R, \tag{5}$$

where α denotes the strength of the user vs item side information.

In order to produce better quality results, apart from the side information, we can also incorporate a collaborative filtering component, which will take into account only the rating information. Different models can be used to include the rating information, but we will use the SLIM model [11]

which estimates the rating that a user would give to an item as a sparse aggregation of the items rated by the user, described in Equation 1. Then by combining Equations 1 and 5, we get:

$$\tilde{R} = \gamma RS + \alpha RIFI^T + (1 - \alpha - \gamma)UWU^T R, \quad (6)$$

where γ is also a parameter.

For a more complete model, we can also include the linear side information. In [12], it was shown that the best method is a collective approach in which it is assumed that there exist correlations between users' copurchase behaviors on two items and the similarity of those items' side information. Thus, the weight matrix S that represents the weights on the ratings should also represent the weights on the features of the side info. In other words, S should satisfy $\tilde{R} = RS$, as well as $\tilde{I} = IS$.

Finally, in order to tie all the pieces together and in order to get the best possible results, since we took into account the cross-feature interactions, apart from the linear part, we can also take into account the higher-order rating information, apart from the linear part. Then, Equation 6 combined with Equation 2 becomes:

$$\tilde{R} = \gamma(RS + R'S') + \alpha RIFI^T + (1 - \alpha - \gamma)UWU^T R. \quad (7)$$

4. INTRA-SET DEPENDENCIES

Top-N recommender systems provide a list of N items, in which the user will be likely interested in. The items that are recommended are the ones that have the highest scores. However, this does not guarantee that the list as a whole is the best possible, even though the individual items are [2,9]. The reason is that the individual items could be very similar. For example, if the item with the highest score for a user is the book 'Alice in Wonderland' and there are two different versions of this book, a paperback and a hard one, then normally these two different versions have very high scores, so there is a high probability that they will end up both being recommended to the user, resulting in the user losing another meaningful recommendation.

This happens a lot, in many recommender systems. One example is music playlists, when the same artist's songs are recommended again and again, or even worse all the versions of the same song are played. Another example is groceries recommendation, when let's say a user is recommended both yellow and green bananas. Finally, another example is the advertisements in a commercial break, that are played in a row.

We propose to stop recommending every item in the list independently but instead follow a set-recommendation approach that will take into account the intra-set dependencies. In order to do this, we will use the side information of the items. For every item, we will find its k-nearest neighbors, in terms of side information features. Very similar items share a lot of the same intrinsic properties, that are encoded in side information. A very high similarity value threshold with the item in question will be imposed. If no neighbors exceeding this high threshold are found for an item, this is acceptable, as we would not like a system that probably eliminates items of importance to the user.

More analytically, top-N recommendation methods create an ordered list of all items for every user. This list is ordered according to how likely a user is to buy this item. The top-N items of this list are provided to the user. In our proposed method, for the first item in the top-N list, the k nearest neighbors of that item are removed from the set of potential candidate items recommended to the user. If the top-N list contained none of these neighbors, nothing changes. If the top-N list contained one or more of these neighbors, then the overall list of items is shifted one or more positions respectively, resulting in a different set of N items being on the top, thus having a new top-N list. The same procedure is done for the second, the third until the Nth item in the list. In the end, the top-N list provided to the user will provide only high-quality recommendations, without two or more items in the list being extremely similar.

The existing methods [18] try to tackle the above problem, in training time, having the users create collections. However, in the vast majority of systems, this is not the default user behavior. Users usually like or buy some items, without creating explicit collections. We believe that it is important in order to address the problem, not to modify user behavior. In order to do this, we will change the prediction part of the model, and not the learning part.

Other existing methods [1, 2] focus on diversity or creating non-obvious recommendations. However, our primary goal is the recommendation quality, so we are totally fine if the recommendations made are not the most unexpected or diverse to a user. In the trade-off diversity vs accuracy, they choose diversity, by making sure that the accuracy does not fall down a prespecified threshold. However, we focus on accuracy. Also, by eliminating only items that it is really crucial to be removed, in order to avoid a repetitive list, we manage to present additional items, that might be of interest to a user, thus potentially even improving the accuracy. Also, in [1], the approach presented will be tried in the context of a massive open online course, but we will try it on a wide variety of datasets that may have different characteristics from an online course. For example, in an online course, there is the notion of prerequisites that will be taken into account, but in other datasets like groceries, this notion does not exist, so we should provide good-quality, diverse recommendations without the help of this notion.

5. CONCLUSION

In this work, we wish to evolve recommender systems to a stage in which the methods developed will not rely on oversimplifications, such as linearity and independence. These assumptions are commmonly made in the methods developed in recommender systems. Based on these assumptions, we can get recommendation results that are general and in some cases sufficient.

However, in a lot of real-world datasets, these assumptions do not hold, resulting in the methods relying on those to underperform. In order to improve the recommendation quality, we think that we should make full use of the different data we have available and learn the more complex relations that might characterize them. The ultimate goal is to develop more sophisticated and realistic models that will improve the recommendation quality.

Towards this direction, in this extended abstract, we suggest three different ways in which this can be achieved. First, we suggest taking into account the higher-order relations in users' rating data. The existing recommender systems compute pairwise associations. However, in real life, purchasing a subset of items in a set might significantly increase the likelihood of purchasing the rest (e.g. in a grocery store users tend to buy items that form the ingredients in recipes). Also, we suggest finding the cross-feature interactions, as many approaches for incorporating side information exploit only linear relations between the features. Finally, we propose developing set recommendation algorithms, that will take into account the intra-set dependencies. In this way, we will predict a set of items such that the overall quality of the set will not just be the sum of the quality values of its members.

6. ACKNOWLEDGMENT

This work was supported in part by NSF (IOS-0820730, IIS-0905220, OCI-1048018, CNS-1162405, and IIS-1247632) and the Digital Technology Center at the University of Minnesota. Access to research and computing facilities was provided by the Digital Technology Center and the Minnesota Supercomputing Institute.

7. REFERENCES

[1] Panagiotis Adamopoulos. Beyond rating prediction accuracy: on new perspectives in recommender systems. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 459–462. ACM, 2013.

[2] Gediminas Adomavicius and YoungOk Kwon. Improving aggregate recommendation diversity using ranking-based techniques. *Knowledge and Data Engineering, IEEE Transactions on*, 24(5):896–911, 2012.

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

[4] Deepak Agarwal and Bee-Chung Chen. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 19–28. ACM, 2009.

[5] Fabio Aiolli. A preliminary study on a recommender system for the million songs dataset challenge. *PREFERENCE LEARNING: PROBLEMS AND APPLICATIONS IN AI*, page 1, 2012.

[6] Evangelia Christakopoulou and George Karypis. Hoslim: Higher-order sparse linear method for top-n recommender systems. In *Advances in Knowledge Discovery and Data Mining*, pages 38–49. Springer, 2014.

[7] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 39–46. ACM, 2010.

[8] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

[9] Derek L Hansen and Jennifer Golbeck. Mixing it up: recommending collections of items. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1217–1226. ACM, 2009.

[10] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. IEEE, 2008.

[11] Xia Ning and George Karypis. Slim: Sparse linear methods for top-n recommender systems. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 497–506. IEEE, 2011.

[12] Xia Ning and George Karypis. Sparse linear methods with side information for top-n recommendations. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 155–162. ACM, 2012.

[13] Rong Pan, Yunhong Zhou, Bin Cao, Nathan Nan Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. One-class collaborative filtering. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 502–511. IEEE, 2008.

[14] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461. AUAI Press, 2009.

[15] Steffen Rendle and Lars Schmidt-Thieme. Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 251–258. ACM, 2008.

[16] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798. ACM, 2007.

[17] Ajit P Singh and Geoffrey J Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 650–658. ACM, 2008.

[18] Greg Walsh and Jennifer Golbeck. Curator: a game with a purpose for collection recommendation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2079–2082. ACM, 2010.

[19] Shuang-Hong Yang, Bo Long, Alex Smola, Narayanan Sadagopan, Zhaohui Zheng, and Hongyuan Zha. Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the 20th international conference on World wide web*, pages 537–546. ACM, 2011.