Methods for Boosting Recommender Systems

Rubi Boim

(Under the supervision of Prof. Tova Milo)
School of Computer Science
Tel-Aviv University
boim@post.tau.ac.il

***Abstract*—Online shopping has grown rapidly over the past few years. Besides the convenience of shopping directly from ones home, an important advantage of e-commerce is the great variety of items that online stores offer. However, with such a large number of items, it becomes harder for vendors to determine which items are more relevant for a given user. Recommender Systems are programs that attempt to assist in such scenarios by presenting the user a small subset of items which she is likely to find interesting. We consider in this work a popular class of such systems that are based on Collaborative Filtering (CF for short). CF is the process of predicting user ratings to items based on previous ratings of (similar) users to (similar) items.**

The objective of this research is to develop new algorithms and methods for boosting CF based Recommender Systems. Specifically, we focus on the following four challenges: (1) improving the *quality* of the predictions that such systems provide; (2) devising new methods for *choosing* the recommended items to be presented to the users; (3) improving the *efficiency* of CF algorithms and related data structures; (4) incorporating recommendation algorithms in *different application* domains.

I. INTRODUCTION

The popularity of online shopping has rapidly grown over the last few years. As the number of items that each online store offers to its consumers can be excessively large (more than 100,000 items in some cases), assisting users in identifying items of interest is crucial. Indeed, much research has been recently devoted to the development of *Recommender Systems* [1], namely programs that attempt to present to the user a small subset of items (out of a much larger items set), which she is likely to find interesting.

Modern Recommender Systems attempt to generate a *personalized* set of recommendations to each user. Such systems often predict a rating (e.g., a grade on a scale of 1 to 5) that a user would assign to an (unseen) item, had it seen it. They consider items with a high predicted rating to be more relevant and assemble the recommendations out of such items. Recommender Systems are often divided into two classes: (1) *Content-Based*, where recommendations are based on semantic properties (preferences) of the items (users), and (2) *Collaborative-Based*, where recommendations are based on previous ratings of (similar) users to (similar) items, with the assumption that users who agreed in the past on item ratings are likely to agree again in the future. For example, if a large portion of users bought both a flat-screen TV and a DVD player, it would be wise to offer the latter for users who have yet purchased it, but did purchased the former. Collaborative-based systems, also known as *Collaborative Filtering* (CF for short), are often considered superior for several reasons: their recommendations tend to be more accurate, they work without the presence of semantic information (which is hard to obtain and maintain in large scale systems) and they more easily support dynamic changes in data/users. Indeed, many of the major e-commerce stores (Amazon, eBay, etc.) exploit CF systems. *Such CF systems are also the focus of this paper.*

This work was partly supported by the EU project Mancoosi, by the Israel Science Foundation and by the US-Israel Binational Science Foundation.

Recommender System, in general, and CF ones in particular, are complex systems and their design naturally introduces a wide range of challenges. Indeed, they triggered extensive research in the past few years (see e.g. [1], [2]). While significant advance has been achieved, there is naturally still room for much improvement and innovations. Specifically, this work focus on the following four challenges: (1) improving the *quality* of the predictions that such systems provide; (2) devising new methods for *choosing* the recommended items to be presented to the users (given that the space available for presentation is limited by the size of the screen); (3) improving the *efficiency* of CF algorithms and related data structures; (4) incorporating recommendation algorithms in *different application* domains, such as Search Engines and Crowdsourcing.

Our research so far focused on the first two challenges, namely improvement of rating prediction and of results presentation. These issues are presented in Section II ([3], [4]) and III ([5], [6]) respectively. We note that some initial results regarding efficiency were considered in our work on the previous challenges. We intended to explore further directions, as well as investigate the forth challenge (different application) in our future research, presented in Section IV. Finally we conclude with a brief review of related work in Section V.

II. IMPROVING RECOMMENDATIONS

Previous works on CF Recommender systems focused mostly on CF performed by a *single* organization over its *own* customer ratings. In contrast, we argue that a *multi-organization* collaboration, even for organizations in distinct subject domains, can greatly improve the quality of the recommendations that the individual organizations provide to their users. To illustrate things, consider the following example.

Assume that `Netflix` - an online movie rental service, `Blockbuster` - a chain of movie rental stores, and `Toys``R''Us` - a toy store chain, wish to collaborate in order to improve the service to their customers. Naturally, each organization has its own database of user ratings and can use it to generate recommendations. But the quality of recommendations may be greatly improved by taking into account information available in the other collaborating organizations. This seems obvious for the case of `Netflix` and `Blockbuster` for which the domain items are similar: Correlations between users interest in one movie store may naturally be used to refine recommendations in the other. But correlations between inter-domain items may also exist and can be leveraged: We may discover, for example, that a large portion of the users who viewed (and liked) "Star Wars" on `Netflix` also bought (and liked) a space-ship model at `Toys``R''Us`, and thus recommend this toy to similar viewers that have not purchased it yet.

While companies specializing in similar domains may be reluctant to cooperate/share data (being competitors), collaboration between companies specializing in distinct, possibly complimentary, domains is rather natural and beneficial to all parties. To substantiate this claim, we developed C2F (Collaborative, Collaborative Filtering) [4], a recommender system that retains the simplicity and efficiency of classical CF, while allowing distinct organizations to collaborate and boost their recommendations. Note that a naive solution that accumulates all data sets into one centralized location (then applies classical CF) is typically not feasible due to the excessive amounts of (constantly updated) data and the independence of the organization. Instead, C2F employs CF in a distributed fashion that maximizes the quality of the generated recommendations, while reducing the amount of data exchanged between the collaborating parties. Key ingredient of the solution are succinct signatures that can be computed locally for items (users) in a given organization and suffice for identifying similar items (users) in the collaborating organizations. C2F employs two main algorithms for computing such signatures, inspired, resp., by works on Dimension Reduction [7] and Features Selection [8]. The former uses a recently developed algorithm based on the Fast Johnson-Lindenstrauss Transform (FJLT) [7]. The latter attempts to reduce the amount of data used using a PTIME greedy heuristic algorithm (we can show the problem to be NP-Complete). For full details see [3].

C2F 's main screen shows the top recommendations identified for the given user. For each recommendation, the user may view its justification: the items which the user liked and which the system believes to be correlated with the recommended item. As C2F leverage information from both local and remote organizations, the items are grouped together by their origin.

An example for such justification is depicted in Figure 1. This specific screenshot is taken from our experiments, where we used data sets within the movies domain, each "specializing" in a distinct area (action movies, documentary movies, etc.) [3], [9]. In this specific example, the user asked for recommendations for action movies. One of the top recommendations was the recent James Bond movie, "Casino Royale". The detailed justification is given to its right. The first row consists of supporting (local) action movies, while the following lines consists of other types of supporting movies, e.g. animated (from other collaborating organizations).

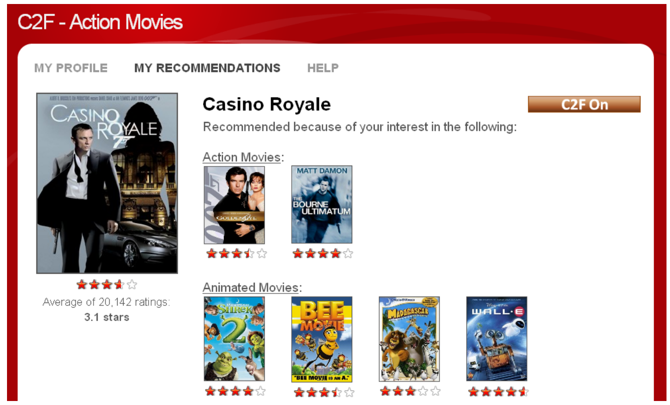

Fig. 1. C2F user interface

We stress that our aim here is not to invent yet another CF algorithm, but rather to present a generic novel technique that allows one to better exploit existing CF algorithms in a distributed, multi-organization setting. Indeed, while our implementation uses specific neighborhood metric/rate aggregation functions to generate recommendations, the technique that we propose can similarly be used for other functions.

III. Diversified Recommendations

Recommender Systems predict the rating that a user would assign to an unseen item, and consider items with a high predicted rating to be relevant. But, which of these highly rated items should be presented first to the user? A naive solution would be to simply sort the items by their estimated rating and present the top-k that fit onto the screen. This however may result in an over-specialized items list. Consider, for instance, the following example: suppose that a user is interested in movie recommendations. Assume that only 5 movies may fit onto the screen and that the top-5 ranked movies, for this user, all happen to be Star Wars sequels. While the given user may indeed like this series, a more *diverse* (and wider) view of the highly ranked movies may be desirable. For instance one that includes a Star Wars movie, but also other movies like Star Trek or E.T. (with the access to more Star Wars movies enabled via a "more of that" zoom-in button).

We developed the DiRec [5] plug-in that provides precisely such a diversification and zoom-in facility. To support this, DiRec has to address two main challenges. The first challenge is how to measure the similarity/diversity of two given items. Previous proposals are typically based on the assumption that some semantic information on items (e.g. the genre of the movie, the director, the actors) is given. In practice, however, many recommender systems do not carry such semantic information [1]. But even when they do, it is not always clear how to define item diversity based on the given semantic information [10]. For example, some movies of the same director/leading actor may indeed be similar, whereas others may not. To overcome this difficulty, DiRec takes a different approach, inspired by work on CF [2]. Instead of relying on semantic information, it defines item similarity (and correspondingly diversity) based solely on ratings that previous users gave to the items.

The second challenge is the need to balance, when choosing items, between two possibly conflicting objectives: presenting highest ranked items vs. choosing highly diverse ones. Previous works attempted to resolve this by assigning a weight to each objective and selecting the item set that maximizes the weighted sum. But the question is which weights to choose?[11]. Here again DiRec resolves the problem by taking an alternative novel approach that avoids the use of weights altogether. We introduce the notion of *priority-medoids*, an adaptation of the classical notion of *medoids*[12] to a context where items have priorities (ratings). *Priority-medoids* ([6]) allow for natural clustering of items and the selection of cluster representatives that balance rank and diversity. While we can show that identifying the best priority-medoids is NP-hard [6], we developed a heuristics based on *priority cover-trees* (an adaptation of the classical cover-trees [13] to our context) which is used by DiRec and provides satisfactory results along with fast response time.

DiRec is designed as a plug-in that can be deployed on CF-based recommender systems, by implementing a simple API, to diversify the recommendations presented to users. Alongside each presented item, DiRec provides a "more of that" zoom-in button that allows to view similar recommended items. Here again, no semantic information is required to identify the similar items, and they are once more presented in an as diversified as possible manner. (Users can then, again, zoom-in on each of presented items, and so on). An interesting property of our implementation is that the priority cover-tree constructed for the initial recommendations set contains most of the information required to support such zoom-in, thus only very minimal further computational effort is required [6].

Figure 2 shows an example within the movie domain for the zoom-in facility: In this specific, the user is initially presented with 5 different movies (the top line). The user then clicks on the "more of that" button of the "Star Wars II: Attack of the Clones" movie. DiRec successfully identifies the Star Wars sequels and presents in response three additional sequels. It also presents two additional movies that do not belong to this series, yet are related, and were chosen by DiRec to provide a more diverse set of recommendations.

Our experimental study examines the operation of DiRec in the context of a movie recommendations system using real data from Netflix [9]. This data set provides only raw user ratings to movies (such as 1 to 5 stars given by individual users) and does not hold any semantic properties besides the movie names. While our algorithms (and so DiRec) use no semantic information, to evaluate the quality of our results we used the movie titles as well as information obtained from IMDb [14], to identify movie sequels and multiple episodes of the same TV program. This allows us to empirically demonstrate that movie sequels and TV program episodes are indeed naturally grouped together by our algorithms, represented by a single item on the screen, and are effectively retrieved, when desired, via the zoom-in mechanism. This is particularly interesting

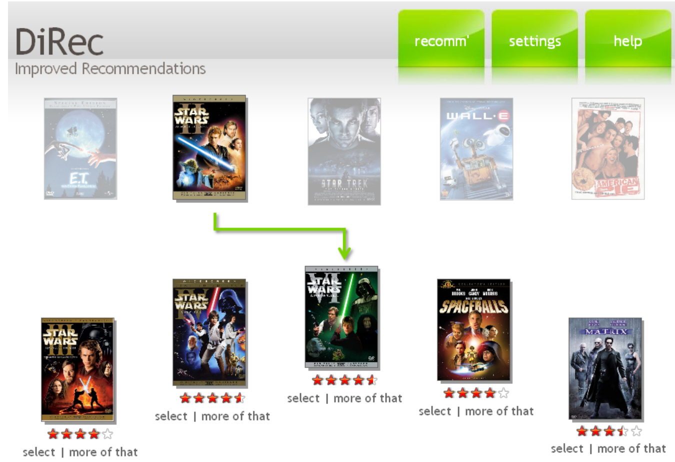

Fig. 2. DiRec "more of that" (zoom-in) mechanism

given the fact that no semantic information (e.g. movie names, actors, etc.) is given to DiRec . We further more examined the quality of our results by a user study of over 50 volunteers that judged themselves the results produced by DiRec . For full details of the experiments see [6].

IV. Future Work

We first consider possible extensions to the obtained results, then consider the two remaining challenges mentioned in the Introduction, namely *efficiency* and *different application*.

Extensions. In Section II, we argued that a multi-organization collaboration, even for organizations operating in different subject domains, can greatly improve the quality of the recommendations that the individual organizations provide to their users. Extending our approach to a Content-based context that may also exploit information about item semantics is a challenging research direction. In such scenarios, for instance, we could quantify (based on semantic knowledge) the degree of similarity of the different collaborating parties, which could be used for communication reduction and predictions boosting.

In Section III, we devised a method that allows CF Recommender Systems to diversify the recommendations that they present to users. Here again, combining our ratings-based (quantitative) approach with a semantic (qualitative) one, when such semantic information is available, is an intriguing research challenge. In such scenarios, we could generalize the diversity measure to captures the behaviors of the users, the semantics of the items, and possibly even the users' current context. Moreover, we could refine the priority-medoid computation and use the available semantic knowledge to further improve the approximation.

Efficiency. Recommender System algorithms, specifically ones based CF, requires extensive amount of computations. Moreover, they do not scale linearly with the size of users / items, causing the optimization factor to be crucial in the upcoming years. A "byproduct" of our work in [3] is the speed up of the computation by over 50% in distributed environments. It is thus interesting to devise an alternative method for *dividing locally the original (large) data set*, one which do not assume the presence of semantic information, and then apply the algorithms in [3] to produce better predictions.

Different Application. We believe that several real-world application domains may benefit from using CF-based concepts and algorithms. *Crowdsourcing* is a recent emerging field in computer science, where the people solve hard computational tasks by playing online games. For instance, a user may be presented with a picture and then asked to describe it with as many tags as possible. The creators of the game gets, in response, a semantic meta-data information regarding each image. Incorporating collaborative-based algorithms may help the creators to match better between the images (or more generally the questions/computational problem that she would like to solve) and the possible users, for increase the quality of the produced results. Other applications, such as *Search Engines*, may benefit as well: In recent years search engines attempt to improve their results by creating a more *personalized* sets of results. For instance, Bing support a new feature which adds information from Facebook (such as the popular "like" tag) to better sort the results of a search query. Incorporating CF-based methods with the search results, such as for instance "People who search this query ultimately clicked this link", may be an exciting research direction.

V. Related Work

We conclude with a brief review of related work, highlighting the relative contributions of our results so far.

Recommender Systems. In this work we focus on a popular class of such systems ([1]) that are based on CF ([2]). Much of the recent research in this field has focused on improving the rating estimations provided by the system [15], [16]. In particular, the Netflix Prize competition [9], challenged researcher around the world to improve these predictions. It was eventually won by a team that employed sophisticated rating aggregation, based on the time that has passed since users placed their ratings [16]. Nevertheless, much improvement can still be made; in this work, we showed how to improve the recommendations generated on one site by leveraging the information of other (different) sites. It is important to stress that sophisticated recommendations algorithms, such as [16], could be applied on each site and still get boosted via our distributed collaboration.

Distributed CF. Previous work on CF in a distributed setting focused on P2P architecture, typically aiming to speed up the computation. A common solution is to decentralize the P2P network w.r.t the users (items), maintaining a "buddies" table at each pier, pointing to the closest users (items) which are believed to share the same taste (be similar) [17], [18]. In all these works the network architecture differs fundamentally from our setting: they consider network of thousands computers, each holding an assigned small part of the data, (useless by itself), whereas we target a much smaller set of collaborating organizations (servers), each holding an entire data set from its corresponding domain.

Closest to our work is [19] that considers the aggregation of rate predictions from multiple sites. The focus however on similar domains with common (user,item) pairs, whereas our solution allows for *inter-domain* collaboration.

Diversification. In [20], [11], the authors also suggest to measure similarity between items using user ratings and attempt to balance between items ratings and diversity. However, they do not provide any formal notion of an optimal such balance and only present heuristics that use predefined thresholds to bound the allowed similarity between items and the drop in rank. In contrast, our priority-medoids admit natural notion of optimality and eliminate the need to use such thresholds/weights altogether. They also allow to support a natural zoom-in process, not addressed by these previous works.

Database Queries. Most relevant to our work, although also targeted to structured data, is [21], where the authors used the notion of (regular) medoids [12] to cluster together the tuples in the query results and to select diversified representatives. They furthermore use (regular) cover-trees to efficiently approximate the optimal medoids/clusters. However, they do not take tuples rating/priority into consideration. An exception is the work in [22] that does take tuples ranking into consideration, but the focus there is on structured data.

References

[1] G. Adomavicius and A. Tuzhilin, "Towards the next generation of recommender systems," *IEEE TKDE*, 2005.
[2] X. Su and T. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, 2009.
[3] R. Boim, H. Kaplan, T. Milo, and R. Rubinfeld, "Improved recommendations via (more) collaboration," *WebDB*, 2010.
[4] ——, "Boosting recommender systems with c2f," *Submitted*, 2010.
[5] R. Boim, T. Milo, and S. Novgorodov, "Direc: Diversifed recommendations for semantic-less collaborative filtering," *ICDE*, 2011.
[6] ——, "Diversifed recommendations for semantic-less collaborative filtering," *Submitted, http://www.cs.tau.ac.il/~boim/publications/divers.pdf*.
[7] N. Ailon and B. Chazelle, "Faster dimension reduction," *Commun. ACM*, vol. 53, no. 2, 2010.
[8] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. of Machine Learning Research*, vol. 3, 2003.
[9] J. Bennet and S. Lanning, "The netflix prize," *KDD Cup*, 2007.
[10] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," *WWW*, 2005.
[11] C. Yu, L. Lakshmanan, and S. Amer-Yahia, "It takes variety to make a world: Diversification in recommender systems," *EDBT*, 2009.
[12] L. Kaufman and P. Rousseeuw, "Finding groups in data: An introduction to cluster analysis," *Wiley's Series in Probability and Statistics*, 1990.
[13] A. Beygelzimer, S. Kakade, and J. Langford, "Cover trees for nearest neighbor," *ICML*, 2006.
[14] "Imdb interface," *http://www.imdb.com/interfaces/*.
[15] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *TKDD*, 2010.
[16] ——, "Collaborative filtering with temporal dynamics," *KDD*, 2009.
[17] A. E. H. Byeong Man Kim, Qing Li, "A decentralized cf approach based on cooperative agents," *WWW*, 2006.
[18] J. Wang, M. J. Reinders, R. L. Lagendijk, and J. Pouwelse, "Self-organizing distributed collaborative filtering," *SIGIR*, 2005.
[19] S. Berkvosky, T. Kuflik, and F. Ricci, "Distributed collaborative filtering with domain specialization," *RecSys*, 2007.
[20] C. Yu, L. Lakshmanan, and S. Amer-Yahia, "Recommendation diversification using explanations," *ICDE*, 2009.
[21] B. Liu and H. Jagadish, "Using trees to depict a forest," *VLDB*, 2009.
[22] J. Stoyanovich, S. Amer-Yahia, and T. Milo, "Making interval-based clustering rank-aware," *Submitted*, 2010.