

Not by Search Alone: How Recommendations Complement Search Results

Daria Dzyabura & Alex Tuzhilin

Stern School of Business, New York University
{ddzyabur,atuzhili}@stern.nyu.edu

ABSTRACT

This paper presents a novel approach to combining search and recommendations methods into one integrated system to satisfy user information seeking needs. It is shown theoretically and experimentally using simulations that the proposed combined approach outperforms “pure” search and “pure” recommendations in those cases when search is hindered by the user’s inability to come up with a complete set of search criteria, and recommendation engine produces mediocre results.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

Keywords: recommender systems, search, hybrid systems.

1. INTRODUCTION

Several prominent recommender systems, such as Netflix and Amazon, support not only recommendations, but also search. For example, 75% of downloads on Netflix come from the recommendation engine and 25% from search [2]. This suggests that neither search nor recommendations alone fully satisfy information seeking needs of these firms’ customers and, therefore, the presence of both of them is required. However, recommendations and search are kept separate on Netflix and Amazon without integration of the two into one information seeking mechanism. In this paper, we explore this integration idea, propose one specific approach of how to combine the two mechanisms into one hybrid model, and demonstrate the advantages of the proposed method.

The concept of integration of search and recommendations has been explored before. For example, [7] argues for the need of convergence of search, recommendations and advertising mechanisms into one common information seeking model. Motivations for this type of convergence come from the following considerations. If a user knows what to search for, then search engines would produce good recall results, i.e., would retrieve all or most of the items of interest to the user. However, if the user cannot specify all the relevant search criteria, and only focuses on a fraction of desired attributes [3], then the recall results can be poor since many interesting items will not be retrieved. In these cases, supplementing search engines with recommendations may greatly improve search. Recommendations not only introduce desirable, previously unconsidered items into the consideration set, but also expose the users to new attributes that they can

subsequently use in their searches. One of the key advantages of recommender systems over search is that they produce “global” recommendations across the whole item space, thus allowing users to discover serendipitous items which they would not think about searching. However, the recommender system’s rating estimates can be noisy, resulting in mediocre recommendations. Users then prefer to manually generate search queries rather than rely only on mediocre recommendations. When neither system works perfectly, a combined system will better satisfy users’ information seeking needs, as argued in [7] and attested by the presence of both technologies at Netflix, Amazon and others.

Although [7] argues for the convergence of search and recommendations, the paper presents only a conceptual framework and the need for such a convergence. It does not provide any specific mechanisms of how to combine these two different technologies and does not explore under what circumstances convergence is beneficial. Moreover, some form of combination of search and recommendations has been explored within the framework of flexible, constrained and critique-based recommendations [1, 4, 5, 6, 8, 9]. The key idea behind all of these methods is that recommendations are provided using not the entire item space, but only some subset of items satisfying various types of constraints and queries. These constraints and queries are specified using different mechanisms across various methods. The underlying idea behind these methods is to limit the space of items considered by the recommender system. This is directly related to search, which also finds objects specified by the search query. However, unlike the ideas described in [7] and explored in this paper, those recommendation constraints and queries serve as inputs for recommender systems, making the constraints and queries *tightly integrated* into the recommender system itself. Therefore, all the methods described in [1, 4, 5, 6, 8, 9] can be viewed as a *tight coupling* between search and recommender systems, which is different from the *loose coupling* approach advocated in [7] and in this paper.

In this paper, we show that it is indeed useful to combine search and recommendations in a loosely coupled manner in some cases in order to achieve higher levels of recall (and even the F-measure). We also propose a specific mechanism for doing so and show that it produces better performance results in some cases. In order to demonstrate this, we first present the specific models of search and recommendations, then describe a particular hybrid model combining the two mechanisms in a loosely coupled manner, and then demonstrate theoretically and via simulations that this hybrid approach dominates pure search and pure recommendations in some settings.

2. THE MODELS OF SEARCH AND RECOMMENDATIONS

In this paper we focus on specific models of items, search and recommendations. Although we have chosen these very specific models in the paper, we believe that our findings are more general

and can be extended to various other models. We plan to work on these extensions in the future, as discussed in Section 5.

2.1. The Items Model. We assume that there is a collection of items I , such as books, songs or movies, and that each item has a set of attributes, such as the list of actors, the director and the year of the movie. Further, we assume that all the attributes of all the items come from the universal collection of attributes A , such as all the actors and movie directors in the movie industry. In our model, we do not consider the particular types of these attributes, only their set A . Finally, we assume that each item is characterized by a collection of attributes from A associated with that item, such as the list of actors and the director of a movie. Furthermore, each attribute can be associated with more than one item, e.g., an actor can appear in more than one movie. This means that, in addition to the collection of items I and attributes A , there is also a bipartite graph G specifying which attributes are associated with which items. This bipartite graph can be connected or disconnected, meaning that it can have one or several connected components. An example of such a graph is presented in Figure 1.

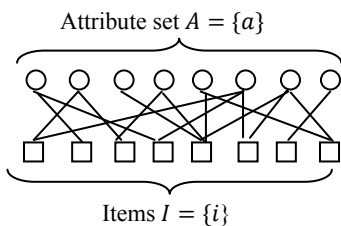


Figure 1: An example of the bipartite graph G .

Let A_i be the set of attributes connected to item i . As is standard in economics and business literature [10], we assume that each item’s utility is the sum of the utilities of its attributes:

$$r_{ij} = \sum_{a \in A_i} x_{aj} \quad (1)$$

where x_{aj} is user j ’s utility for attribute a , and r_{ij} is the overall utility (rating) of item i for user j . An item is considered to be “good” if its utility is above a threshold value α .

2.2. Search (S). Given the bipartite graph presented in Section 2.1, the search process is performed as follows in our model S . First, the user selects a single attribute a from A and submits the search query using this attribute. The search engine returns all the items associated with attribute a in graph G . The user examines all the returned items, identifies those that are of interest to him/her and identifies the criteria for the next search based on this examination of the selected items. Since we have made a simplifying assumption that the search query uses only a single attribute, this means that the user needs to chose only one such attribute for the next search. Although restrictive, we believe that the single attribute simplifying assumption can be lifted in future work and does not restrict the findings of this paper in significant ways.

In this paper, we model this iterative search process and the user’s searching behavior as follows. First, we assume that the user selects all the items from the set of items returned by search on attribute a that have utility above the threshold α and “consumes” those items (e.g. adds the selected Netflix movies to the queue). The user then selects the next search attribute probabilistically in our user model from those that occurred in the products that s/he liked from the search results, with probability proportional to the number of times the attribute occurred. For example, if the user searched for “James Bond” at time 1, and one of the films he liked

was Casino Royale, he may choose to search for Daniel Craig at time 2. If the search produced n desirable items with a certain feature, the user would be n times more likely to search for that feature in the next period, as if it had occurred only once.

This process of *repeatedly* selecting items and attributes in our model continues until saturation when the point of diminishing returns or a steady state is reached and no more new good items are produced for two periods in a row.

2.3. Recommendations (R). In this paper, we do not consider any specific method of estimating unknown ratings. We only assume that the attribute utilities x'_{aj} are estimated somehow using any of the existing techniques proposed in the literature [11] (note that if we know estimates x'_{aj} , then we can compute the ratings r_{ij} using equation (1)). Also, we assume that these estimations, x'_{aj} , are noisy but correlated with their true values x_{aj} and can be modeled as draws from a bivariate normal distribution:

$$\begin{bmatrix} x_{aj} \\ x'_{aj} \end{bmatrix} \sim \mathcal{N}(\mu, \Sigma),$$

where

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

The parameter ρ is crucial because it determines quality of recommendations: larger values of parameter ρ lead to more accurate and lower values to less accurate estimations of unknown ratings. Note that this model R is general because a wide range of existing recommender systems can fit into this generic model.

A “good” item – to be recommended – is one that has utility higher than the threshold value α described in Section 2.1.

2.4. Combining Search and Recommendations into a Hybrid R+S Model. The search S and recommendation R models can be combined into a single R+S model in several different ways. In this paper, we focus on one particular hybrid R+S method when certain recommendations are added to the search results. More specifically, we generate a search list using the method described in Section 2.2, produce recommendations as described in Section 2.3, and combine them as follows. When the user selects the attribute for the next search criterion, s/he can choose *either* from the search results, *or* the product recommendations. This way, product recommendations may not only introduce the recommended products into the consideration set, but also influence the *future* search path. For example, a user searching for action films may, if recommended one Hitchcock horror film, search for more Hitchcock and other similar films in the next periods, now that Hitchcock and horror are salient attributes. Understanding attribute salience is critical in connecting user-driven search and recommendation systems: attributes that are made salient to the user by product recommendations can, if the user chooses, be used as search criteria in the future.

3. THE PROBLEM FORMULATION

We want to compare the performance of the hybrid approach with pure recommendations and pure search and demonstrate that in some settings the hybrid approach dominates pure search and pure recommendations. We start with this comparison theoretically and show that there exist some instances of pure search and pure recommendation models where the hybrid approach outperforms them in terms of the recall measure. Then in the next section we demonstrate this further via simulations. We proceed with our theoretical explorations by formulating and proving the following propositions. They compare performance of the hybrid and pure models only in terms of recall since discovery of all the good

items was the main motivation of this work. We provide comparisons of precision in the simulation experiments.

Proposition 1. There exists a set of items I , a set of attributes of these items A , and a bipartite graph G of attribute-item features, such that for any starting search query q , pure search (S) retrieves weakly fewer “good” items than the combined R+S model.

Sketch of Proof. Assume that set I consists of N items and these items collectively have N attributes (forming set A). Further, graph G connects each attribute to a single item bijectively, thus producing N clusters, each cluster having only one item. Then any search query q retrieves only 1 item, and no subsequent query generated based on the retrieved list can improve this result. Therefore, the final outcome is that one item is retrieved for any query q if this item is “good”, and no items retrieved if it is “bad”. In contrast, if we add recommendations to the pure search case, then, if the recommended product turns out to be good, the user will end up with extra good item received. ■

Proposition 2. There exists a set of items I , a set of attributes of these items A and a bipartite graph G of attribute-item features, such that for any recommendation r generated on graph G and for any search query q , the combined R+S method generates at most as many “good” recommendations as the pure S model.

Sketch of Proof. Assume that graph G is fully connected, i.e., every item is connected to every attribute. It is clear that any search query q will eventually find *all* the good items using the search process (described in Section 2.2) and, therefore, recommendations are superfluous for such graph G . Therefore, the combined R+S model performs at most as well as the S model. ■

Proposition 2 says that the pure search model is so good on the fully connected graph G that any additional recommendations added to the search model are superfluous because the sequence of iterative item retrievals launched by search query q will eventually find all the good items in the fully connected graph G .

These two propositions theoretically show that the R+S model dominates the pure S model for the recall measure for *some* bipartite graphs. However, the proofs rely on extreme cases (it is unlikely that graph G is either fully connected or disconnected in practice). Thus, it is essential to explore what happens for the whole range of configurations of graph G , as well as compare R+S and R models, which we do in Section 4 using simulations.

4. COMPARING THE R+S MODEL WITH PURE R & S CASES USING SIMULATIONS

We simulate the S, R and the R+S approaches as follows. First, we assume that graph G has 10,000 items in I , 5,000 attributes in A . To generate the edges, we first randomly partition items and attributes into 50 groups, each group representing closely related items. We generate an edge between any item and attribute *within* a group with probability P_w ; we generate edges between attributes and items *not* in one group (*across* groups) with probability P_a . In our experiments, we use $P_w = 0.1$ and $P_a \in [0.001, 0.01]$.

Second, we simulate the search process S. We begin with query q having a single attribute a and then iterate through the subsequent searches using the mechanism described in Section 2.2. We stop these iterations when no new “good” items are retrieved after $k = 2$ iterations. The “goodness” of an item (i.e., how valuable it is to the user) is defined with the utility parameter α that takes value of 3.5 in our simulations. Third, we simulate recommendations as described in Section 2.3. We vary the value of parameter ρ from 0 to 1. Finally, we simulate the hybrid R+S model as described in

Section 2.4 by doing search and adding recommendations produced by the recommender system.

Based on these simulated data, we compare the performance of the R+S model to that of R and S models for a range of parameter values. The most important parameters that crucially affect the performance results are P_a and ρ . P_a determines the connectivity and “clusteredness” of the graph, and thus controls the quality of search; ρ determines the quality of recommendations.

Fig. 1(a,b) present the average recall measure for the R, S and R+S simulations while varying P_a and ρ , respectively and keeping the other parameter constant. The R+S model outperforms the R and S models in the region where R is weak ($\rho < 0.7$). Interesting interactions occur in the region of $P_a=0.003$, and $\rho=0.6$, where the respective curves cross. We now focus on this region of P_a and ρ .

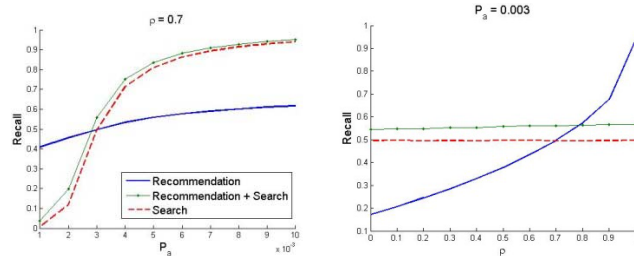


Fig 1a: Recall varying P_a

Fig 1b: Recall varying ρ

Fig. 2 presents the probability density functions of the recall measure for the R, S and R+S models computed across 10,000 users for parameter values $P_a = 0.003$ and $\rho = 0.6$. The hybrid R+S model outperforms the pure S and pure R models for these values of parameters. The performance improvement is statistically significant at the highest level. The same result holds for the parameters $P_a = 0.01$ and $\rho = 0.6$ for the recall measure, as shown in Fig. 3, also with a high level of significance. However, this is not true for the case of $P_a = 0.001$ and $\rho = 0.6$, as shown in Fig. 4, where pure R model significantly outperforms both the S and R+S models. This is because when the P_a value is so low the product space is very clustered, which hinders the search process: the user is likely to search only one cluster and stop. Therefore the recall values for the S and R+S cases are diminished, but the pure R system does not suffer from this limitation.

We conclude from these results that the hybrid model R+S does not always outperform the S and R models across all the parameters. Performance improvements for the R+S vs. the R and S cases happen when both search and recommendation results are only mediocre and there is not a big difference in their performance, as in Fig. 2. If either search or recommendations drastically dominates the other, then the hybrid model does not outperform *both* pure search and recommendation cases, as seen in Fig 3 and 4. It is this “middle ground” case of both models performing reasonably well and one model marginally outperforming the other (as in the $P_a = 0.003$ and $\rho = 0.6$ case of Fig. 2) when the R+S model really helps, as Fig. 2 demonstrates.

So far, we have compared the R+S and the R and S models only in terms of the recall measure, which is important, but does not give the full picture of the R+S vs. R and S comparisons. Therefore, Fig. 5 presents the comparison of these models in terms of precision. The pure R model outperforms the R+S and S models in terms of precision in case of parameters $P_a = 0.003$ and $\rho = 0.6$. This is because parameter $\rho = 0.6$ produces fairly precise recommendations, whereas search produces a long and

“complete” but not a precise list of outcomes, resulting in low precision values. R also outperforms R+S on the F-measure, but

we do not plot it due to space constraints.

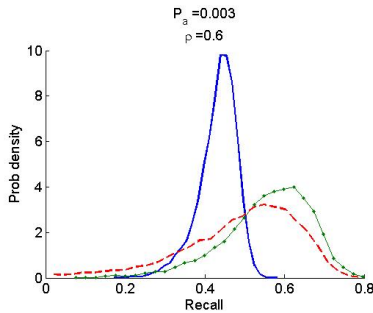


Fig 2: Recall for $P_a=0.003, \rho=0.6$

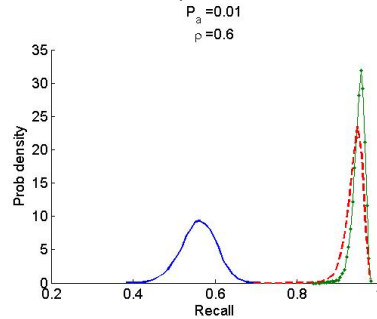


Fig 3: Recall for $P_a=0.01, \rho=0.6$

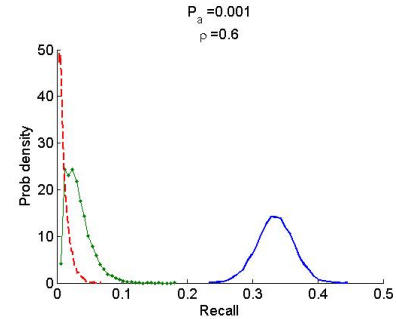


Fig 4: Recall: $P_a=0.001, \rho=0.6$

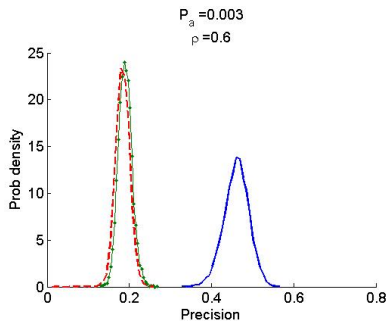


Fig 5: Precision: $P_a=0.003, \rho=0.6$

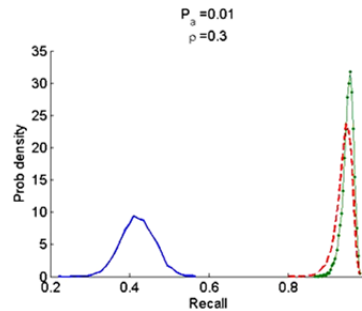


Fig 6: Recall: $P_a=0.01, \rho=0.3$

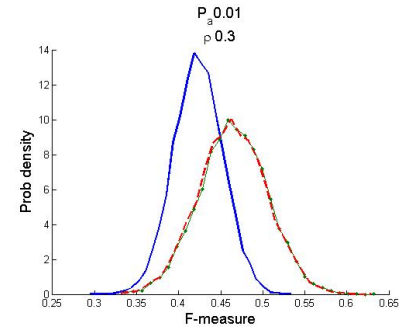
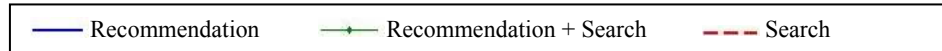


Fig 7: F-measure: $P_a=0.01, \rho=0.3$



To further explore this phenomenon, we considered less accurate recommendations ($\rho = 0.3, P_a = 0.01$). The results for recall and the F-measure are presented in Figs 6 and 7, respectively. The R+S model outperforms the R model for $P_a = 0.003$ and $\rho = 0.3$ both in terms of recall and the F-measure. Regarding the R+S vs. S case, the R+S model slightly outperforms the S model in terms of recall and achieves practically the same performance results as S in terms of the F-measure. This means that the R+S model performs better or similar than the pure S and R models in terms of not only the recall measure but also the F-measure for some parameter settings.

5. CONCLUSIONS

In this paper, we studied the problem of integrating search and recommendations in a loosely coupled system that combines the search and recommendation results. We proposed a specific method of combining the two approaches and showed that the proposed hybrid R+S model outperforms pure search S and pure recommendation R models in terms of the recall and the F-measures in certain parameter settings, but does not for other settings.

This paper presents our preliminary investigations of this important and complex problem. In the future, we plan to conduct experiments on real data and users to investigate if similar results hold on real data. We also plan to study further under which conditions the hybrid approach dominates the pure cases, ideally, trying to formulate the necessary and/or sufficient conditions for such dominance. We also plan to study various other methods of loose coupling of the search and

recommendation methods besides the specific one presented in this paper.

6. REFERENCES

- [1] Adomavicius, G., Tuzhilin, A. and Zheng, R. REQUEST: A Query Language for Customizing Recommendations. Information Systems Research, 22(1), 2011.
- [2] Amatriain, X and Basilico, J. Netflix Recommendations: Beyond the 5 Stars (Part 1), April 2012.
- [3] Dzyabura, Daria. The Role of Changing Utility in Product Search. NYU Stern Working Paper, 2013.
- [4] Burke, R. Knowledge-based recommender systems, Encyclopedia of Library and Information Sciences, 69(32), 2000.
- [5] Chen, L. and Pu, P. Critiquing-based recommenders: survey and emerging trends. User Modeling and User-Adapted Interactions, 22(1-2), 2012.
- [6] Felfernig, A., Friedrich, G., Jannach, D., and Zanker, M. Developing Constraint-based Recommenders. Recommender Systems Handbook, Ch. 6, 2011.
- [7] Garcia-Molina, H., Koutrika, G., and A. Parameswaran. Information Seeking: Convergence of Search, Recommendations, and Advertising, CACM, Nov. 2011.
- [8] Koutrika, G., Ikeda, G., Bercovitz, B. and Garcia-Molina, H. Flexible recommendations over rich data. RecSys 2008.
- [9] McCarthy, K., Reilly, J., McGinty, L. and Smyth, B. On the Dynamic Generation of Compound Critiques in Conversational Recommender Systems. Adaptive Hypermedia, 2004.
- [10] McFadden, D. Econometric Models for Probabilistic Choice Among Products. Journal of Business, 53(3), 1980.
- [11] Ricci, F., Rokach, L., Shapira, B. and Kantor, P. Recommender Systems Handbook, Springer, 2011.