Poster: Toward the Development of Richer Properties for Recommender Systems

David Shriver
University of Nebraska-Lincoln
Lincoln, NE, USA
dshriver@cse.unl.edu

ABSTRACT

The performance of recommender systems is commonly characterized by metrics such as precision and recall. However, these metrics can only provide a coarse characterization of the system, as they offer limited intuition and insights on potential system anomalies, and may fail to provide a developer with an understanding of the strengths and weaknesses of a recommendation algorithm. In this work, we start to describe a model of recommender systems that defines a space of properties. We begin exploring this space by defining templates that relate to the properties of coverage and diversity, and we demonstrate how instantiated characteristics offer complementary insights to precision and recall.

1 INTRODUCTION

Recommender systems aid users in making decisions when lacking personal experience or knowledge [6] or when the set of choices is overwhelmingly large [4]. They can be seen everywhere, from e-commerce to news recommendation. However, standard metrics used by developers to validate the performance of recommender systems, such as precision and recall, are limited in several ways.

First, they provide little information or intuition to the developer or provider of a recommender system about the quality and usefulness of the recommendations the system provides. For instance, in this work, we show that a known hybrid recommendation algorithm has precision and recall superior to a known model-based collaborative filtering algorithm when evaluated on the MovieLens dataset, yet, the content-based algorithm achieves those gains by providing excessively conservative recommendations. The model-based collaborative filtering algorithm recommends 3.5 times more unique items and is able to produce a unique ranked list for almost every user. Such insights are lost with coarse summary statistics like precision and recall.

Second, when the number of known relevant items is very small relative to the number of items available to be recommended, recommender systems are likely to exhibit extremely low precision and recall values. Therefore, a system developer will have little intuition as to whether a precision of 0.001% should be considered supremely satisfying or deeply troubling.

ICSE '18 Companion, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5663-3/18/05.
https://doi.org/10.1145/3183440.3195082

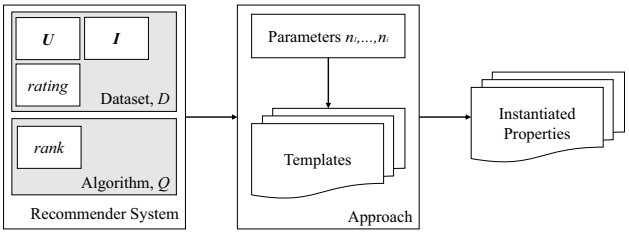

Figure 1: Our approach instantiates the parameters of property templates based on users, items, ratings, and ranks.

Third, not only do precision and recall values for an individual system convey little intrinsic intuition about the system, conclusions about the relative superiority of one system over other systems based on precision and recall values can be highly misleading if the recommendations provided by the putatively superior system are anomalous or counterintuitive.

In this work, we delineate an approach for defining properties of recommender systems that can provide more intuitive characterizations of the nature and quality of the recommendations produced by a recommender system. Using our approach, we define two property templates relating to the *coverage* and *diversity* of recommender systems, and explore the intuitions they can provide when applied to recommender systems. Armed with such properties, developers will be able to validate richer recommendation behaviors complementary to accuracy metrics such as precision and recall.

2 DERIVING RECOMMENDER PROPERTIES

To simplify the definition of recommender system properties, we begin by defining a simple model for recommender systems. For our model, let $\mathcal{U}$ be a finite set of users, $\mathcal{I}$ a finite set of items, and $\mathcal{R}$ a finite, ordered set of rating values. A dataset D defines the partial function $rating\colon \mathcal{U} \times \mathcal{I} \to \mathcal{R}$ that captures how users rate items.

Given D and a parameter k for computing top-k rankings, a recommendation algorithm Q computes a partial function $rank_{k,Q}\colon \mathcal{I} \times \mathcal{U} \to [1,k]$ for $k \leq |\mathcal{I}|$. In what follows, we drop the subscripts Q and k from $rank_{k,Q}$, since Q is typically apparent from the context, and k is a parameter of Q.

With this model, we can specify relatively simple templates—based on users, items, ratings, and rankings—that can be instantiated to capture a broad set of usefulness properties about a recommender system, as shown in Figure 1. In general, a template is defined as a function mapping a recommender system and a set of parameters to a Boolean value. Two such property templates are described below.

The first property is the number of recommended items (NRI), which provides insight on the coverage of the recommender system, the proportion of items that a recommender system can recommend [1, 3, 8]. The NRI property can be defined as: $\#\{i \mid i \in \mathcal{I} \wedge \exists u \in \mathcal{U} : rank(i,u) \neq \bot\} \geq n$, where the function # counts the number of elements in the set. The parameter n is instantiated as the minimum value which makes the property evaluate to true. A higher value of n means that a higher number of items were recommended, thus the recommender has higher coverage.

The second property is the number of recommendation sets (NRS), which provides a view of the diversity and personalization of the recommender system. Diversity is a measure of the dissimilarity of recommendations and can refer to the personalization or uniqueness of users recommendation lists [9]. Higher diversity between sets of recommendation are often preferred because it implies that the recommender produces more personal recommendations. The NRS property can be defined as: $\#\{\{i \mid i \in \mathcal{I} \wedge rank(i,u) \leq k\} \mid u \in \mathcal{U}\} \geq n$. The parameter n is instantiated as the minimum value which makes the property true, and is equal to the number of recommendation sets produced by the recommender. If the instantiated value is high, then the rankings are likely to be more personalized.

Our approach treats usefulness properties as logical properties of a recommender system, rather than as metrics. This representation enables instantiated templates to to represent a space of values for a property, providing a richer characterization of the behavior than a coarse, singular value metric. Additionally, logical properties can be used as assertions when evaluating a recommender system with an evolving dataset, enabling developers to ensure that the behavior is consistent over time.

3 INITIAL EXPLORATION

We instantiated two property templates on two recommender algorithms, Item-Item and LightFM, using the MovieLens 20M dataset [2]. The Item-Item algorithm is a model-based collaborative filtering algorithm in which similarity scores are computed between pairs of item rating vectors, and predicted ratings are computed by aggregating the ratings of the most similar items [7]. The LightFM algorithm is a hybrid recommender algorithm that uses both rating values, as well as item attributes to build a recommender model [5].

Table 1: Precision, Recall, and Instantiated Property Values for Each Recommender System

	Precision	Recall	NRI	NRS
Item-Item	0.00179	0.000341	2791	138417
LightFM	0.00520	0.00213	786	85784

The instantiated values for each property are shown in Table 1, along with the precision and recall values for each algorithm. Observing only the precision and recall values, one might conclude that LightFM performs better than the Item-Item algorithm, due to LightFM having higher values for both precision and recall. However, the NRI property shows that the Item-Item recommender recommends a higher proportion of the items, which may be desirable if a developer wants to ensure higher utilization of the item set. The Item-Item recommender also produces more recommendation sets, providing a unique set for almost all 138493 users. The value for NRS indicates that Item-Item produces more personalized sets of recommendations than the LightFM recommender on the MovieLens dataset.

4 CONCLUSIONS AND FUTURE WORK

In this work, we have presented a simple model for recommender systems that can be used to define property templates. Instantiated property templates provide developers with a richer, more intuitive view of recommendation behavior during validation of system performance. For example, they can be used as assertions for validating behavior as the dataset evolves over time.

In our initial exploration, we have shown how two simple properties defined using our approach provide insights into the coverage and diversity behavior of recommender systems, complementary to precision and recall. The properties presented here represent only a small sample of the defined space and were selected to show the power of our approach to represent existing usefulness properties.

In future work, we will perform a more exhaustive exploration of this space as it may reveal additional useful properties for recommender systems. For instance, using our approach, we can define characteristics to calculate user influence scores based purely on the user, items, ratings, and rankings, without requiring a subset of the data to be used as a test set or repeated modification of the dataset as are required by existing techniques. Longer term, we want to use the characteristics to assist in the explanation of certain recommendations that do not meet a developer's expectations.

ACKNOWLEDGMENTS

I would like to thank Sebastian Elbaum, David S. Rosenblum, and Matt Dwyer for their discussion and guidance. This work has been supported in part by National Science Foundation award #1526652.

REFERENCES

[1] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. 2010. Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity. In *Proceedings of the Fourth ACM Conference on Recommender Systems*. ACM, New York, NY, USA, 257–260.

[2] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (Dec. 2015), 19 pages.

[3] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. 1999. An Algorithmic Framework for Performing Collaborative Filtering. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, 230–237.

[4] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating Collaborative Filtering Recommender Systems. *ACM Trans. Inf. Syst.* 22, 1 (Jan. 2004), 5–53.

[5] Maciej Kula. 2015. Metadata Embeddings for User and Item Cold-start Recommendations. In *Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems co-located with 9th ACM Conference on Recommender Systems (RecSys 2015), Vienna, Austria, September 16-20, 2015.*, Toine Bogers and Marijn Koolen (Eds.), Vol. 1448. CEUR-WS.org, 14–21.

[6] Paul Resnick and Hal R. Varian. 1997. Recommender Systems. *Commun. ACM* 40, 3 (March 1997), 56–58.

[7] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the International Conference on World Wide Web*. ACM, New York, NY, USA, 285–295.

[8] Guy Shani and Asela Gunawardana. 2011. *Evaluating Recommendation Systems*. Springer US, Boston, MA, 257–297.

[9] Tao Zhou, ZoltÃąn Kuscsik, Jian-Guo Liu, MatÃžÅą Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang. 2010. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences* 107, 10 (2010), 4511–4515.