Recommender Systems Evaluation: A 3D Benchmark

Alan Said
TU Berlin
alan@dai-lab.de

Domonkos Tikk
Gravity R&D
domonkos.tikk@gravityrd.com

Yue Shi
TU-Delft
y.shi@tudeft.nl

Martha Larson
TU-Delft
m.a.larson@tudelft.nl

Klara Stumpf
Gravity R&D
klara@gravityrd.com

Paolo Cremonesi
Politecnico di Milano
paolo.cremonesi@polimi.it

ABSTRACT

Recommender systems add value to vast content resources by matching users with items of interest. In recent years, immense progress has been made in recommendation techniques. The evaluation of these has however not been matched and is threatening to impede the further development of recommender systems. In this paper we propose an approach that addresses this impasse by formulating a novel evaluation concept adopting aspects from recommender systems research and industry. Our model can express the quality of a recommender algorithm from three perspectives, the end consumer (user), the service provider and the vendor (business and technique for both). We review current benchmarking activities and point out their shortcomings, which are addressed by our model. We also explain how our 3D benchmarking framework would apply to a specific use case.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval - Retrieval models

1. INTRODUCTION & MOTIVATION

Recommender systems identify items suitable for specific users in large content collections. Despite recent commercial and research efforts, a systematic evaluation model that addresses and considers all aspects and participants of the recommender system is still missing. In this paper we propose a *3D Recommender System Benchmarking Model* that covers all dimensions that impact the effectiveness of recommender systems in real-world settings. The concept builds on a study of benchmarking settings from research and industry and provides a common comparison of recommender systems, independent of setting, data and purpose. Our benchmarking concept captures three evaluation aspects which are shared in all recommender systems, independent of whether they are research systems or industrial products. As three main evaluation dimensions we identify *user requirements*, *business requirements* and *technological constraints*, each represented by a set of qualities which ensure the general applicability of these procedures. For each particular recommendation problem, the instantiation and relevance of these requirements should be specified.

The motivation behind this framework is the growing importance of recommender systems. Users cannot be assumed to have the necessary overview to specify their information needs in vast content collections. However, with a variety of data and the recommendation task, the comparison of algorithms, approaches and general concepts becomes infeasible due to the inherent differences in requirements, design choices, etc. This calls for a comprehensive benchmarking framework that sets data- and task-specific requirements driven by particular real-world applications.

The benefits of benchmarking. Benchmarks formulate standardized tasks making it possible to compare the performance of algorithms. They have been highly successful in the areas of information retrieval, e.g. Text Retrieval Conference (TREC) [12] and the multimedia retrieval ImageCLEF [7], TRECVid [9] and MediaEval [6]. Benchmarks yield two types of benefits; (1) they serve to support the development of new technologies in the research community [9, 11] and (2) they create economic impact by bringing research closer to the market [8].

Existing recommendation benchmarks. Today's benchmarks are limited by their simplified views of users and of data. The problem setting of the Netflix Prize[1], groundbreaking at its time, was focused on a single *functional requirement*: the qualitative assessment of recommendation was simplified to the root mean squared error of predicted ratings. Its simplified view treated users as needing no further output from the recommender system than a rating on individual items. The data set was equally restricted to user ratings, additional information available in a real-world recommender system environment were not considered. Furthermore, the Prize did not take *non-functional requirements* into account, which arise from business goals and technical parameters of the recommendation service, though aspects as *scalability*, *reactivity*, *robustness* and *adaptability* are key for the productive operation of recommender systems.

The series of context-aware movie recommendation (CAMRa) challenges explored the usefulness of contextual data in recommendations. The 2010 challenge[2] provided special features on the movie mood, movie location, and intended audience (Moviepilot track), as well as social relationship between users and user activities on a movie-related social site (Filmtipset track). The time of the recommendation was also considered as context (Week track). Although the challenges expanded the data sources used, the evaluation translated real-world user needs into the classification accuracy metrics to evaluate the system in the contest, and non-functional requirements of the solutions were not investigated.

The limitations of the Netflix Prize and CAMRa series are characteristics of currently existing benchmarks and data sets. The concept presented in this paper approaches this

Workshop on Recommendation Utility Evaluation: Beyond RMSE (RUE 2012), held in conjunction with ACM RecSys 2012 September 9, 2012, Dublin, Ireland

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

[2] http://www.dai-labor.de/camra2010/challenge/

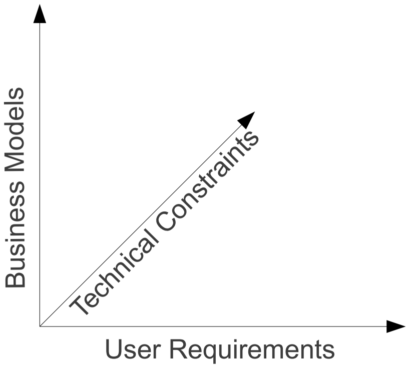

Figure 1: The three proposed evaluation axes.

challenge by placing central focus on real-world user needs; large, heterogeneous, multi-source data sets and evaluating both *functional* (quality-related) and *non-functional* (technical and business goals-related) requirements.

2. 3D RECOMMENDATION EVALUATION

In order to extend the state of the art of evaluation, we propose a concept for evaluation metrics that incorporates the needs from all perspectives in the recommendation spectrum. The concept defines a set of benchmarking techniques that select the correct combination of (*i*) data sets, (*ii*) evaluation methods and (*iii*) metrics according to a three dimensional requirement space: *business models*, *user requirements* and *technical constraints*, see Fig. 1.

Business models allow a company to generate revenue. Different models lead to different requirements in terms of the expected value from a recommender system. For instance, in a pay-per-view video-on-demand business model, the goal of the recommender system is to increase sales to allow the company to maximize revenues. However, in subscriber-based video-on-demand business models, the driving forces may be to get users to return to the service in the future (a typical showcase where recommender systems help [1]). Business models may be influenced by the choice of the objective function in the recommender algorithm; prediction-based or ranking-based functions reflect different business metrics.

User requirements reflect users' perspectives. Recommenders are assets for user satisfaction and persuasion, i.e., they try to influence a user's attitude or behavior [4], the usability of the systems affect the user's perception of the system. Recommendations may have different goals, e.g. reduce information overload, facilitate search, and find interesting items increasing the quality and decreasing the time of the decision-making process.

Technical constraints. Recommender systems in real-life must take into account a number of technical requirements and constraints. These can be classified as *data* and *system constraints*, *scalability* and *robustness* requirements. **Data constraints** relate to the service architecture, e.g. satellite TV lacks a return channel for feedback, hindering the use of collaborative filtering algorithms. **System constraints** derive from hardware and/or software limitations, e.g. in a mobile TV scenario, the processing power in the hand-held device is limited; excluding resource-heavy algorithms on the client side. **Scalability requirements** derive from the need of instant recommendations to all users on all items. These requirements are particularly strict in linear TV, where viewers are used to quick responsiveness. **Robustness requirements** are needed to create good services, able to work in case of data or component failure in distributed systems.

3. EVALUATION SETTING

3.1 Current evaluation methodologies

Existing evaluation methods for recommender systems can be classified into *system-oriented evaluation*, *user-oriented evaluation* or a combination of both [3].

In **system-oriented evaluation** (*off-line*) users are not involved in the evaluation, instead, a data set is partitioned into training and test sets. Using the training set, data points in the test set are predicted. In **user-oriented evaluation** (*on-line*) feedback from users interacting with the system is collected by explicit questions or implicit observing.

Competitions and challenges built around recommender systems are mostly organized to find the most accurate models. As described in Table 1, recommender systems are mostly evaluated off-line and often, the business value of the technologies is not examined. Even though the accuracy may influence user satisfaction and revenue increase indirectly, there exists no way to evaluate the dimensions of user requirements and business models. In most of the cases, the off-line evaluation scheme is chosen. Algorithms are often evaluated by error, ranking or classification accuracy measures. Many challenges (e.g. Netflix Prize) use explicit ratings to profile users, other recommender scenarios (e.g. item-2-item recommendation) are not addressed. Technical constraints are uncommon in contests, the exception being the RecLab Prize[3]. If a certain method performs well on a data set, the integrability in a real-world system is still not addressed. This deficiency is partially solved by online testing methods (as seen in CAMRa): recommender systems were tested in a real environment, but an objective metric to show the real applicability of the tested system is missing. In the RecLab Prize, the evaluated metric is revenue increase generated by the system. The organizers also specified non-functional requirements to be eligible for the semi-final (top 10 teams), but user requirements are not considered. These approaches all contain metrics and methods moving towards our 3D model, but none of them provide a comprehensive model.

3.2 Currently existing metrics

On-line evaluation is the only technique able to measure the true user satisfaction; conducting such evaluations is however time consuming, and cannot be generally applied, rather only to limited scenarios [2]. Contrary, off-line testing has the advantage to be immediate, and easy to perform on several data sets with multiple algorithms. The question is whether differences between the off-line performance of algorithms can be carried over to differentiate their online performance in various recommendation situations.

Classification metrics measure how well a system is able to classify items correctly, e.g. precision and recall. **Predictive metrics** measure to what extent a system can predict ratings of users. As rated items have an order, predictive accuracy metrics can be used to measure the item ranking ability. **Coverage metrics** measure the percentage of items for which the system can make recommendations [13]. **Confidence metrics** measure how certain the system is of the accuracy of the recommendations. Additionally, many recommender systems algorithms use **learning rate metrics** in order to gradually increase quality.

A recommender system can recommend accurate items, have good coverage and diversity and still not satisfy a user, if they are trivial [10]. The state-of-the-art of the evaluation metrics of recommendation reflects different recommendation tasks. **Diversity**, **novelty**, **serendipity** and **user**

[3] `http://overstockreclabprize.com/`

Table 1: An overview of some recommender system-related contests from the perspective of our 3D evaluation

Challenge	Task(s)	Metric	Mode	User	Business	Technical
Netflix Prize	minimize rating prediction error	RMSE	off-line	indirect: error measure	not addressed	not addressed
KDD-Cup'07	**1**: predict who rated what **2**: predict number of ratings	RMSE	off-line	not addressed	detect trends & popular items	not addressed
RecLab Prize	Increase revenue	revenue lift	online & off-line	not addressed	revenue lift	response/learning time, scalability
KDD-Cup'11	minimize rating prediction error split popular/unpopular items	RMSE ErrorRate	off-line	indirect: error measure find interesting or irrelevant items	not addressed	not addressed
KDD-Cup'12	prediction followed users click trough rate prediction	MAP@3 MAE, AUC	off-line	exploring interesting users & sources	not addressed ad targeting (CTR)	not addressed
CAMRa'10	context-aware; **1**: temporal, **2**: emotional, **3**: social	MAP, P@N, AUC	off-line & online	contextual information influences preference	not addressed	not addressed
CAMRa'11	group recommendation rater identification	ErrorRate	off-line	group & target recommendation	indirect: satisfaction	not addressed
CAMRa'12	find users for specific items	impact	on-line	split interesting and irrelevant content	increase audience	not addressed

satisfaction are especially difficult to measure off-line. Diversity is important for the usefulness of a recommendation and therefore there is a need to define an intra-list similarity metric [13]. Novelty and serendipity are two dimensions of non-obviousness [3].

3.3 Possible Extensions of Methods & Metrics

Real-world recommender systems should satisfy (*1*) functional requirements that relate to qualitative assessment of recommendations and (*2*) non-functional requirements specified by the technological parameters and business goals of the service. Functional and non-functional requirements should be evaluated together: without the ability to provide accurate recommendations, no recommender system can be valuable. As poor quality has adverse effects on customers, it will not serve the business goal. Similarly, if the recommender does not scale with a service, not being able to provide recommendation in real time, neither users nor service provider benefit from it. Thus, a trade-off between these requirements is needed for an impartial and comprehensive evaluation of real-world recommenders. Scalable recommenders provide good quality recommendations independently of the data size, growth and dynamic. They are able to (*1*) process huge volumes of data during initialization using computation resources linearly scalable with data size; and (*2*) serve large amounts of parallel recommendation requests in real time without significant degradation in service quality. In our model, scalability is found on the technical requirement axis.

Reactivity ensures good recommendations in real-time where the time threshold depends on the use case, typically in the range of 10–1000 ms. Adaptability is important to react for changes in user preferences, content availability and contextual parameters. In our 3D model, reactivity and adaptability belong to the user requirement axis.

Robustness is needed to handle partial, missing or corrupted data both in the system initialization and operational phases. Robustness belongs to the business axis of our model.

Generally speaking, none of the requirements are mutually exclusive, instead, optimization should be based on a combination of them – adapted for the setting in which the recommender system will be deployed [5].

This example of a Video-on-Demand (VoD) service from the IPTV industry serves as a potential scenario for our model. Business goals include increased VoD sales and customer retention, but may have additional aspects (promoting content). The technical constraints are partly specified by the middleware and the hardware/software configuration of the service provider, these all influence the response time of the service which is crucial. Via the service interface, the user gets recommendations based on the context, which might be translated into different recommendation tasks. From a user perspective, easy content exploration and context dependent recommendation may be the most important aspects.

4. CONCLUSION

We proposed a 3D Recommender System Benchmarking model that extends the state-of-the-art and addresses both functional and non-functional real-word application-driven aspects of recommender systems. Following the proposed concept, the benchmarking activities within the community will encompass the full range of other recommender system use cases and algorithmic approaches. The comprehensive evaluation methodology will boost the development of more effective recommender systems, and make it possible to focus research resources productively and for industry technology providers to increase the uptake of recommender technology.

5. REFERENCES

[1] M. B. Dias, D. Locher, M. Li, W. El-Deredy, and P. J. Lisboa. The value of personalised recommender systems to e-business: a case study. In *RecSys '08*. ACM, 2008.

[2] M. Gorgoglione, U. Panniello, and A. Tuzhilin. The effect of context-aware recommendations on customer purchasing behavior and trust. In *RecSys '11*, pages 85–92. ACM, 2011.

[3] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1), 2004.

[4] R. Hu. Design and user issues in personality-based recommender systems. In *RecSys '10*. ACM, 2010.

[5] T. Jambor and J. Wang. Optimizing multiple objectives in collaborative filtering. In *RecSys '10*. ACM, 2010.

[6] M. Larson, M. Soleymani, P. Serdyukov, S. Rudinac, C. Wartena, V. Murdock, G. Friedland, R. Ordelman, and G. J. F. Jones. Automatic tagging and geotagging in video collections and communities. In *ICMR '11*. ACM, 2011.

[7] H. Müller. *ImageCLEF experimental evaluation in visual information retrieval.* Springer, Heidelberg, 2010.

[8] B. Rowe, D. Wood, A. Link, and D. Simoni. Economic impact assessment of NIST's text retrieval conference (TREC) program. Technical report, July 2010.

[9] A. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and TRECVid. In *MIR '06*, 2006.

[10] L. Terveen and W. Hill. Beyond recommender systems: Helping people help each other. In *HCI in the New Millennium.* Addison-Wesley, 2001.

[11] T. Tsikrika, J. Kludas, and A. Popescu. Building reliable and reusable test collections for image retrieval: The Wikipedia Task at ImageCLEF. *IEEE Multimedia*, 99(PrePrints), 2012.

[12] E. M. Voorhees. Overview of TREC 2005. In *TREC*, 2005.

[13] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW '05*. ACM, 2005.