A 3D Approach to Recommender System Evaluation

Alan Said[†]
Technische Universität Berlin
alan@dai-lab.de

Brijnesh J. Jain[†]
Technische Universität Berlin
jain@dai-lab.de

Sahin Albayrak
Technische Universität Berlin
sahin@dai-lab.de

[†] Both authors contributed equally to this work

Abstract
In this work we describe an approach at multi-objective recommender system evaluation based on a previously introduced 3D benchmarking model. The benchmarking model takes user-centric, business-centric and technical constraints into consideration in order to provide a means of comparison of recommender algorithms in similar scenarios. We present a comparison of three recommendation algorithms deployed in a user study using this 3D model and compare to standard evaluation methods. The proposed approach simplifies benchmarking of recommender systems and allows for simple multi-objective comparisons.

Author Keywords
Recommender Systems; Evaluation; Benchmarking

ACM Classification Keywords
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval - Information filtering

General Terms
Algorithms, Design, Experimentation, Human Factors

Introduction
Recommender systems are traditionally evaluated through metrics such as precision, recall, root-mean-squared error, etc. [1]. These measure only show one type of performance,

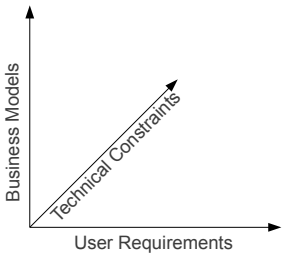

Figure 1: The 3 axis forming the 3D evaluation model.

namely the objective recommendation accuracy, not taking into consideration subjective user-centric values, business aspects or technical constraints [4] – the two latter are largely neglected in recommender systems research [1]. Even in cases where multi-objective evaluation is applied, e.g. [2], the evaluation focuses on recommendation accuracy. In real world scenarios, business- and technology-centered measures are just as important, if not more, than accuracy only. A recent evaluation model incorporating all three values was presented by Said et al. [4]. This model, if applied in the context of a real worl, market-driven recommender system, simplifies the algorithm-to-algorithm comparison of recommendation systems. In order to show the validity of the model, in this paper we apply the model in a system with three different recommendation algorithms and present the results from the comparison.

Evaluation

Evaluation of recommendation algorithms is usually done by measuring aspects such as rating prediction accuracy or rank of known *good* items in offline settings. Offline in this case meaning the training and evaluation of an algorithm based on previously collected interaction data between users and items. Commonly a portion of the data is used for training the algorithm, and a portion for evaluation. This type of evaluation does however only capture one aspect of the algorithm, namely it's prediction quality, i.e. the users predicted response to the algorithm. However, as proposed by Said et al. [4] there are several other aspects of a recommendation algorithm which should be taken into consideration, i.e. the *business model*, the *technical constraint* and the *user requirements*, as shown in Fig. 1. In the context of the figure, traditional evaluation only captures the user requirements axis.

Multi-Objective Evaluation

A comprehensive evaluation of a recommender system f amounts in a multi-objective evaluation. For this, we take into account different evaluation metrics $E_i(f)$ from the three dimensions user, business, and technology. Examples of evaluation metrics are the customer return rate, customer churn rate, average computation time as well as standard evaluation metrics such as the RMSE, precision, and recall.

For the sake of convenience, we assume that all evaluation metrics are formulated as cost functions to be minimized. We define the multi-objective evaluation function E by

$$E(f) = \Big(E_1(f), \ldots, E_p(f)\Big)^T,$$

where E is a vector of cost functions E_i. This setting corresponds to the problem of multi-objective optimization.

The question at issue is as follows: Suppose that we have two recommender systems f and f'. Which recommender system do we prefer on the basis of our multi-objective evaluation function E? The field of multi-objective optimization suggests several approaches to this question. Here, we present one common approach. We combine the evaluation metrics E_i to a single weighted global criterion

$$U(f) = \boldsymbol{w}^T E(f) = \sum_i w_i E_i(f) \qquad (1)$$

where $w_i \geq 0$ are weights that may be interpreted as the importance of evaluation metric E_i. Using the utility function U, we prefer a recommender system f to f', if $U(f) < U(f')$. It is important to note that the choice of evaluation metrics E_i and weights w_i are problem dependent design decisions.

Experiments

The aim of this experiment is to demonstrate multi-objective evaluation of the following three recommender algorithms:

- k-Nearest-Neighbor (kNN): A standard algorithm used for recommendation. kNN recommends items which are liked by users similar to oneself, which can cause low diversity due to effects of popularity, e.g. highly rated popular movies are often recommended to very many users.
- k-Furthest-Neighbor (kFN): An algorithm which turns kNN inside-out and recommends items which are unliked by users dissimilar to oneself. This algorithm is tuned for higher diversity and thus recommends more diverse items.
- Random (Rnd): A random recommender, simply recommending a random selection of the items available.

We collected for each of the three axis the following data: (i) business axis: intention of return; (ii) user axis: usefulness of recommendation; and (iii) technology axis: computation time required to calculate recommendations. For this purpose, we performed a user study ($n = 132$). The study was set up as a simple movie recommender (described in detail in [3]) where users would rate a number of movies and receive recommendations based on the input. We employed the above three recommendation algorithms.

For the business and user axis, we asked users upon receiving a set of 10 recommended movies, whether they would consider using the system again (intention of return), and whether the recommendations were useful (usefulness of recommendation). Answers to the questionnaire amount to ratings normalized to a scale from 0 (not appropriate) to 1 (highly appropriate). Based on these data, we selected the following evaluation metrics:

- $E_b(f)$ measures the average intention of return of f
- $E_u(f)$ measures the average usefulness of f
- $E_t(f)$ measures the utility of the average computation time t_f required by f according to

$$E_t(f) = \frac{a}{1 + \exp\left(t_f/T - 1\right)},$$

where $T = 30$ is the maximum time considered as acceptable, and a is a factor scaling $E_t(f)$ to 1 if $t_f = 0$. The evaluation metric E_t is 1 at $t_f = 0$ and approaches zero with increasing computation time.

According to Eq. (1), we combined the evaluation metrics to a utility function of the form

$$U(f) = w_b \cdot E_b(f) + w_u \cdot E_u(f) + w_t \cdot E_t(f).$$

The choice of $\boldsymbol{w}$ is shown in Table 1.

Results & Discussion

The utility values for different weights $\boldsymbol{w}$ (shown in Table 1) show that kNN attains better values than kFN for two out of three evaluation metrics, E_b, E_u, and ties in one, E_t. As a consequence, the utility value U of kNN is always equal to or better than the one of kFN regardless of how we choose the weights. In multi-objective optimization, we say that kNN is pareto-superior to kFN. As expected, when business- and user-requirements are preferred, kNN and kFN outperform the random recommender. In a use case, where computation time is the most critical constraint, the random recommender outperforms the others solely based on the

speed of the recommendation. In a scenario where all three axis are equally important, kNN performs best.

	kNN	kFN	Rnd
$E_u(f)$	0.53	0.52	0.44
$E_b(f)$	0.56	0.51	0.36
$E_t(f)$	0.80	0.80	0.99
$U(f)$ with $\boldsymbol{w} = (0.\bar{3}, 0.\bar{3}, 0.\bar{3})$	0.63	0.61	0.60
$U(f)$ with $\boldsymbol{w} = (0.6, 0.3, 0.1)$	0.57	0.55	0.47
$U(f)$ with $\boldsymbol{w} = (0.3, 0.6, 0.1)$	0.58	0.54	0.43
$U(f)$ with $\boldsymbol{w} = (0.1, 0.1, 0.8)$	0.75	0.74	0.87

Table 1: Utility values $U(f)$ for different weights $\boldsymbol{w}$. The maximum accepted time is $T = 30s$.

The results illustrate, that an appropriate choice of evaluation metrics, as well as weight parameters, are critical issues for the proper design of a utility function. This design process is highly domain and problem dependent. Once a proper utility function has been set up, the performance of different recommender algorithms can be objectively compared. Since current state-of-the-art-recommenders are often optimized with respect to a single recommendation accuracy metric, introducing multi-objective evaluation functions from the different contexts of a deployed recommender system sets the stage for constructing recommender algorithms that optimize several individual evaluation metrics without at the same time worsening another.

Conclusion & Future Work

Evaluating recommender systems traditionally only considers the users' requirements on a system, in a real-world scenario other factors should be taken into consideration as well. In this paper we evaluate three recommendation algorithms according to a three dimensional evaluation model taking user requirements, business models and technical constraints into considerations when determining the quality of an algorithm. We show how the model can be applied to a system in order to create a contextual quality estimate of the system showing the quality of a recommendation engine from all three perspectives. Prior to benchmarking algorithms, the expected utility in each of the three dimensions should be defined together with their importance, e.g. a system which requires instant answers has a high weight w for $E_t(f)$, a system which requires high accuracy has a high weight w for $E_u(f)$, etc.

To provide a rule-of-thumb for real-world recommendation benchmarking, we are currently studying different aspects of recommendation quality in order to provide a taxonomy of measures and their importance in different contexts of deployment.

References

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

[2] Jambor, T., and Wang, J. Optimizing multiple objectives in collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, ACM (New York, NY, USA, 2010), 55–62.

[3] Said, A., Fields, B., and Jain, B. J. User-centric evaluation of a k-furthest neighbor collaborative filtering recommender algorithm. In *Proceedings of the ACM 2013 conference on Computer Supported Cooperative Work*, ACM (New York, NY, USA, 2013).

[4] Said, A., Tikk, D., Shi, Y., Larson, M., Stumpf, K., and Cremonesi, P. Recommender systems evaluation: A 3d benchmark. In *Proceedings of the Workshop on Recommendation Utility Evaluation: Beyond RMSE (RUE 2012)*, RUE'12, CEUR-WS Vol. 910 (2012).