Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity

Mouzhi Ge
TU Dortmund
44221, Dortmund, Germany
mouzhi.ge@tu-dortmund.de

Carla Delgado-Battenfeld
TU Dortmund
44221, Dortmund Germany
carla.delgado@tu-dortmund.de

Dietmar Jannach
TU Dortmund
44221, Dortmund Germany
dietmar.jannach@tu-dortmund.de

ABSTRACT
When we evaluate the quality of recommender systems (RS), most approaches only focus on the predictive accuracy of these systems. Recent works suggest that beyond accuracy there is a variety of other metrics that should be considered when evaluating a RS. In this paper we focus on two crucial metrics in RS evaluation: coverage and serendipity. Based on a literature review, we first discuss both measurement methods as well as the trade-off between good coverage and serendipity. We then analyze the role of coverage and serendipity as indicators of recommendation quality, present novel ways of how they can be measured and discuss how to interpret the obtained measurements. Overall, we argue that our new ways of measuring these concepts reflect the quality impression perceived by the user in a better way than previous metrics thus leading to enhanced user satisfaction.

Categories and Subject Descriptors
C.4 [**Performance of Systems**]: Measurement techniques.

General Terms
Measurement, Performance, Reliability.

Keywords
Recommender system, evaluation metric, coverage, serendipity.

1. INTRODUCTION
Over the last decade, different recommender systems were developed and used in a variety of domains [1]. The primary goal of recommenders is to provide personalized recommendations so as to improve users' satisfaction. As more and more recommendation techniques are proposed, researchers and practitioners are facing the problem of how to estimate the value of the recommendations. In previous evaluations, most approaches focused only on the accuracy of the generated predictions based, e.g., on the Mean Absolute Error. However, a few recent works argue that accuracy is not the only metric for evaluating recommender systems and that there are other important aspects we need to focus on in future evaluations [4, 8].

The point that the recommender community should move beyond accuracy metrics to evaluate recommenders was for example made in [8]. There, informal arguments were presented supporting that accurate recommendations may sometimes not be the most useful ones to the users, and that evaluation metrics should (1) take into account other factors which impact recommendation quality such as serendipity and (2) be applied to recommendation lists and not on individual items.

This paper analyzes the evaluation of recommender systems focusing on the quality of recommendations rather than only on their predictive accuracy of algorithms. Quality as a concept has been extensively discussed over the last decades and various definitions can be found in a wide range of literature. A definition that is especially prevalent in marketing and the service industries is the one of Gronroos [3]. Gronroos defines quality as meeting and/or exceeding customer's expectations. Another well-known definition was introduced by [6], who defined quality as "fitness for use". From these definitions we are able to conclude that (1) a high-quality recommendations need to be fitting their intended purpose and (2) the actors behind this purpose are the ultimate judges of the quality of recommendations. One particular problem in the RS domain is that the purpose of a RS and the actors behind (costumers and sellers) take different positions and have different perspectives on the purchase act. Thus, a "good" recommendation can be considered at the same time one that makes the customer happy or points him to an interesting item as well as one that maximizes the sales margin.

In this paper, we focus on two crucial evaluation metrics for RS: coverage and serendipity. While coverage concerns the degree to which recommendations cover the set of available items and the degree to which recommendations can be generated to all potential users, serendipity is concerned with the novelty of recommendations and in how far recommendations may positively surprise users. A recommender with high coverage represents to the end user a more detailed and careful investigation of the product space, therefore an indicator of quality. Serendipity on the other hand is supposed to let the system appear more lively by making non-trivial and surprising recommendations. Introducing serendipity should help to reveal unexpressed users' wishes, on one hand ameliorating the tedious task of acquiring users' preferences and on the other hand providing interesting shopping experiences. A feeling like seeing at a display window something you did not know that existed but that perfectly fits your lifestyle. Although both metrics are commonly discussed in the literature [4, 8, 11] and some attempts to evaluate specific systems using them were already made [10], the measurement and trade-off between the two are still unclear. Our intention is to provide new ideas on how coverage and serendipity can be measured and specifically investigate the trade-off between the two metrics.

RecSys'10, September 26–30, 2010, Barcelona, Spain.

This paper is organized as follows. The next two sections respectively discuss possible metrics for measuring coverage and serendipity. Then we discuss the trade-off between them and finally conclude the paper by providing practical implications and future research directions.

2. COVERAGE

According to [4], the coverage of a recommender is a measure of the domain of items over which the system can make recommendations. In the literature, the term coverage was mainly associated with two concepts: (1) the percentage of the items for which the system is able to generate a recommendation, and (2) the percentage of the available items which effectively are ever recommended to a user [4, 11]. Though different authors differ with respect to terminology, here we adopt the definition from [4] and refer to (1) as prediction coverage, and to (2) as catalogue coverage.

Prediction coverage is highly dependent on the recommendation engine and its input. If we consider collaborative filtering (CF) the inputs are item ratings. If we consider knowledge-based recommenders (KBR), the inputs are some sort of means-end logic such as explicit recommendation rules and the user preferences. In both cases, the system will be able to generate recommendations for items for which it received enough input, i.e. ratings in the case of CF and applicable rules in the case of KBR. If we use I to denote the set of available items and I_p to denote the set of items for which a prediction can be made, a basic measurement for prediction coverage can be given by:

$$Prediction\ coverage = \frac{|I_p|}{|I|}$$

The way in which I_p is obtained varies depending on the used recommender technology. To give an example, some collaborative filtering systems are just able to make predictions for items which have more than a fixed number c of ratings assigned. In such a case I_p can be considered as the set of items for which the number of available ratings exceeds c. On the other hand, some knowledge-based systems return lists with a fixed maximum number of items that passed a filter. In this case I_p can be given by the items that are able to pass through at least one of the less restrictive filters allowed by the system.

[4] considered the benefits of refining the prediction coverage measurement to take into account the usefulness of an item in a recommendation. One option would be to favor recommenders that propose items that are beyond the mainstream taste and that appear relatively seldom in recommendation lists ("the long tail"). In addition, recommendations that include items for new users or new items in cold-start situations could be also seen as particularly valuable. Assuming that $r(x)$ gives the usefulness of an item x, we propose to introduce a weighting factor in the calculation of the coverage in order to take the relative utility of items in a recommendation list into account:

$$Weighted\ prediction\ coverage = \frac{\sum_{i \in I_p} r(i)}{\sum_{j \in I} r(j)}$$

We are left with the task to define how to obtain the values of $r(x)$. One option is accuracy, as for many recommenders this is the data that is available. For collaborative filtering, it is usual to remove some of the existing ratings and measure accuracy to what extent the system is able to predict these held-out ratings (even if ratings are user-dependent, an average of the predictive accuracy can be calculated for one item). For knowledge based recommenders accuracy can be measured by monitoring the way the user refines his preferences to browse the items' catalogue. If the user changes his preferences completely after seeing a recommendation, this reveals low interest in the returned items and is an indicator that accuracy of the recommendation was low.

Beside accuracy, also other measures can be used to estimate usefulness. The *novelty* of an item could be another promising one, as this information is in general available for every item. Besides that, new items are the ones that the user has possibly little knowledge about and for which it is important for a recommender to be able to make the user aware of them and promote them. For items the user already has heard a lot about (think of currently top-selling, generally-liked items) even an accurate recommendation would not be too meaningful. This aspect is related to serendipity and will be further discussed in the next section. Yet another possible measure is *effectiveness* which can be approximated by comparing what has been recommended and what has actually been purchased, or the click-through rate.

Let us now consider catalog coverage. Catalog coverage can be a particularly valuable measure for systems that recommend lists of items (e.g. top 10 most-suited items), as this is not taken into account by prediction coverage. Catalog coverage is usually measured on a set of recommendation sessions, for example by examining for a determined period of time the recommendations returned to users [4]. Let us denote I_L^j as the set of items contained in the list L returned by the jth recommendations observed during the measurement time. N shall be used to denote the total number of recommendations observed during the measurement time and let I be the set of all available items (i.e., the catalog). We propose to measure catalog coverage as follows:

$$Catalog\ coverage = \frac{|\bigcup_{j=1\ldots N} I_L^j|}{|I|}$$

It is expected that for low values of N, small increases in N lead to reasonable increases on Catalog coverage. When evaluating a recommender it is then interesting to observe for which value of N the Catalog coverage gets stable, i.e., in which situations it is barely affected when N increases.

We can refine the Catalog coverage measure if for the system under consideration those items that are unsuitable for recommendation should definitely be filtered out to fit the user interests (i.e., systems that should avoid recommending DVDs to people who are searching for CDs). In this case, the respective decrease in coverage should be balanced by the usefulness of the items to the user. In this way, the system is not considered to have poor coverage for leaving out of items that are of no interest to the user. If B^j denotes the set of items that could be considered useful to be returned in recommendation j, we propose to calculate the *Weighted Catalog coverage* as follows:

$$Weighted\ Catalog\ coverage = \frac{|\bigcup_{j=1\ldots N}(I_L^j \cap B^j)|}{\left|\bigcup_{j=1\ldots N} B^j\right|}$$

The idea behind the formula is to consider for each recommendation made during the evaluation time, how many items were indeed useful, i.e. the intersection between the set of recommended items and the set of useful items. The numerator of the formula will provide the total number of recommended items

that were useful during the measurement time. The denominator stands for the total number of useful items at the same time (i.e., the number of items that were worth recommending).

The same discussion regarding usefulness that was presented for prediction coverage applies here. Determining B^n should be done by taking into account metrics of usefulness that apply for each recommender. A very naïve approach is to take B^n as the general category of products the user is interested in, for example, all books. Another approach could be to use the recommendation function itself and let B^n encompass all items for which the recommendation function exceeds a determined threshold.

3. SERENDIPITY

Recently, several researchers began to investigate the aspect of serendipity in the context of recommender systems. In order to understand the nature of this concept, Iaquinta et al. [5] analyzed the etymology of this word. They found that serendipity is mostly related to the quality of recommendations in RS-related research, that it largely depends on subjective characteristics, and that it is a difficult concept to study. Various definitions are proposed for this concept in the recommender system domain. For example, in [4] serendipity is defined as a measure of the extent to which the recommended items are both attractive and surprising to the users. That means a highly serendipitous recommendation would help a user to find a surprising and interesting item. Based on this definition we can see two important aspects related to serendipity. First, a serenedipitous item should be not yet discovered and not be expected by the user; secondly, the item should also be interesting, relevant and useful to the user. Similar definitions are also found in other works. For instance, [11] considered serendipity as a measure of how surprising and successful the recommendations are. In [8], serendipity is defined as the experience of the user who received an unexpected and fortuitous recommendation.

Ideally, when we successfully implement serendipitous encounters in the recommendations, we are able to avoid "obvious" recommendations in collaborative filtering systems [4], solve the over-specification problems in content-based systems [5] and also can help users reveal their unexpected interests [7]. However, as serendipity is judged from a subjective sense, it is also combined with a risk that if the serendipitous recommendations lead the users to an unsatisfying or useless result, the users in the future may stop following the recommendations or using the recommender application as a whole [11].

From our review, we found that experimental studies of serendipity are very rare. Thus it is an open question how to implement serendipitous recommenders. However, from the way we measure serendipity we can infer how to use the measuring determinants to improve serendipity. The metric proposed in this paper is intended to catch the two essential aspects of serendipity, unexpectedness and usefulness. We consider that if a user found an item unexpected and useful, he is surprised and interested in this item.

In order to measure unexpectedness, we need a benchmark model that generates expected recommendations. We therefore follow the approach of [9] and assume that a primitive prediction model shows high ratability and produces low unexpectedness. Consider that when using the same data source, a primitive prediction model and a recommender system respectively generate a set of recommendations. Let PM be a set of recommendations generated by a primitive prediction model and RS denote the recommendations generated by a recommender system. When an element of RS does not belong to PM, we consider the element is an unexpected recommendation. Note that the primitive prediction model may diversify the recommendations according to certain patterns. We therefore calculate the unexpected set of recommendations as follows.

$$UNEXP = \mathrm{RS} \setminus \mathrm{PM}$$

However, unexpected recommendations may be not always useful. Therefore we use *u(RS\PM)* to describe the usefulness of the unexpected recommendations. We define that RS_i is an element in *UNEXP*. When $u(RS_i) = 1$, RS_i is considered to be useful and when $u(RS_i) = 0$, it means RS_i is useless to the user. N is the total number of elements in *UNEXP*. The usefulness of RS_i can be judged by the user. Once we have determined unexpectedness and usefulness, we are able to define serendipity as follows.

$$SRDP = \frac{\sum_{i=1}^{N} u(\mathrm{RS}_i)}{N}$$

Note that this new serendipity metric not only determines the ratio of serendipitous recommendations in the recommendation list but also takes their usefulness into account. For example, a user is particularly interested in action movies. If we add a documentary to the recommendation list, the user might not expect such a movie as it does not comply with his interest. If he watched the documentary and found it interesting, we can consider this documentary as a valuable serendipitous item.

In order to differentiate between novelty, diversity and serendipity, we provide the following example. In the list of recommended action movies, the user might also find an unknown movie which is interesting to him. This action movie is then called novel recommendation instead of serendipitous recommendation because he might discover this movie by himself. If we found that the user likes action movies might also like comedy movies, we can diversify the recommendation list by adding a comedy movie. It is then called a diverse recommendation instead of a serendipitous recommendation as he might be not surprised about the recommendation.

Regarding the question of how we could conduct experiments to validate our metric, let us consider the following setup. We present a list of song recommendations to a user who can listen to each song for up to 30 seconds. The songs are presented without any metadata such as artist name or song title. After listening, the user is asked to provide the feedback for two questions: First, if he already knew the song, which would help us to assess *unexpectedness* of the recommendation. Second, if he liked the song, which can serve as a basis to measure an item's *usefulness*. Thus we could obtain a realistic view of users' perception.

Although a serendipitous recommendation represents an unexpected finding for the user, it does not mean we should recommend a totally unrelated item to the user. It is obviously difficult to recommend a totally appropriate item; however, we are able to reduce the risks involved with serendipity. For example, we can provide a separate list to the user that contains serendipitous recommendations. Thus even if the surprising

recommendations do not match the users' interest in any case, the user may not stop using the recommendation service as a whole. This has been done by Amazon.com (*http://www.amazon.com*). We may reduce the risk by introducing the possibly related categories. The plausibility of category relations can be judged by business rule mining or our life experiences. Thus we might create more chances to improve serendipity.

Although serendipity is combined with some risk, we believe that including serendipitous items will make the recommender system more alive and interesting. In the following section, we will discuss the trade-offs between coverage and serendipity.

4. TRADE OFFS

It is not hard to see that accuracy, serendipity and coverage are closed related and influence one another. As stated at [2], catalog coverage usually decreases as a function of accuracy. The point is that accurate recommendations are directly proportional to the amount of available data the system has to generate recommendations. If we consider collaborative filtering, the more ratings we have available for each item and for each user, the more specific and/or personalized can the generated recommendations be. As usually some popular items receive much more ratings than others, an accurate system will tend to recommend items from this group. If we consider knowledge based recommenders, recommendations are often based on rules that map functional user requirements to technical item characteristics. The more restricted the requirements specifications accepted by the system are, the more specific will be the recommendations, increasing accuracy and sacrificing catalog coverage. For example, imagine that the system allows the user to specify a number P of item properties he is interested in. Then, the system recommends items that are optimal regarding this combination of P properties. Only items that can be among the top X optimal items regarding P will be covered by the recommendations.

Catalog coverage and serendipity are closely related. Not every increase in coverage leads to serendipity. An increase in serendipity will however also lead to higher catalog coverage. Conversely, the more we focus on increasing accuracy by yielding catalog coverage, the less serendipity happens. As stated in [5], even a "perfect" content-based recommender cannot find surprising items. When we attempt to increase the serendipity, we need to enlarge catalog coverage to rarely rated items.

Although increasing serendipity might negatively impact accuracy, this should by no means be seen as a synonym of producing spurious recommendations. The increase of serendipity has to be strategically done in order to alleviate the risk of confusing the users or having a distrust effect. We believe this risk can be alleviated by (1) providing explanations as to why the item is recommended and, most importantly, (2) better arranging the recommendation list or using multi-lists. A recommendation list usually gives the impression that the recommended items are more or less the same and the top one is the most accurate. However if one or more items are noticeably different from the other recommended items, it might stimulate the user's interest and curiosity. In turn it can improve the quality of the recommendation.

5. CONCLUSION

This paper considered two evaluation metrics for recommender systems that go beyond accuracy: coverage and serendipity. These metrics were designed to take the quality and "usefulness" of recommendations into account. We started a discussion about how these two metrics should be interpreted when contrasted with accuracy measures and how they relate to one another. As a particular aspect in the measurement of coverage and serendipity, we proposed to better incorporate the factor of *usefulness* of an item.

Experiments with the usage of these metrics in real scenarios are the next step towards obtaining an extended evaluation framework for recommender systems. This is our envisioned future work. Overall, we see our work as a starting point for the development of new evaluation methods for recommender system that introduce additional perspectives of how the quality of recommendation lists can be estimated.

6. REFERENCES

[1] Adomavicius, G. and Tuzhilin, A. 2005. Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), pp. 734-749.

[2] Good, N., Schafer, J., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J. and Riedl, J. 1999. Combining collaborative filtering with personal agents for better recommendations. Conference of the AAAI, Florida, USA. pp. 439-446.

[3] Gronroos, C. 1983. Strategic management and marketing in the service sector. Marketing Science Institute. USA.

[4] Herlocker J., Konstan J., Terveen L. and Riedl J. 2004. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems, 22(1), pp. 5–53.

[5] Iaquinta L., Gemmis M., Lops P., Semeraro G. 2008. Introducing serendipity in a content-based recommender system. Eighth International Conference on Hybrid Intelligent Systems, Barcelona, Spain. pp 168-174.

[6] Juran, J.M., Gryna, F.M. and Bingham, R.S. 1974. Quality control handbook, 3rd edition, McGraw-Hill, New York, USA.

[7] Kamahara, J., Asakawa, T., Shimojo, S. and Miyahara, H. 2005. A community-based recommendation system to reveal unexpected interests. 11th International Multimedia Modeling Conference, Melbourne, Australia. pp. 433 – 438.

[8] Mcnee S., Riedl J and Konstan J. 2006. Accurate is not always good: How Accuracy Metrics have hurt Recommender Systems, Conference on Human Factors in Computing Systems, Quebec, Canada. pp. 1-5.

[9] Murakami T., Mori K., and Orihara R. 2007. Metrics for evaluating the serendipity of recommendation lists. New frontiers in artificial intelligence: JSAI, pp. 40-46.

[10] Parameswaran, A. G., Koutrika, G., Bercovitz, B., and Garcia-Molina, H. 2010. Recsplorer: recommendation algorithms based on precedence mining. In *Proceedings of SIGMOD'10*. ACM, New York, NY, 87-98.

[11] Shani G. and Gunawardana A. 2009. Evaluating Recommendation Systems. Technical report, No. MSR-TR-2009-159.