Looking for "Good" Recommendations: A Comparative Evaluation of Recommender Systems

Paolo Cremonesi[1], Franca Garzotto[1], Sara Negro[1],
Alessandro Vittorio Papadopoulos[1], and Roberto Turrin[2]

[1] Department of Electronics and Information, Politecnico di Milano,
Via Ponzio, 34/5 - 20133 Milano (Italy)
{cremones,garzotto,negro,papadopoulos}@elet.polimi.it
[2] Moviri srl, R&D

Abstract. A number of researches in the Recommender Systems (RSs) domain suggest that the recommendations that are "best" according to objective metrics are sometimes not the ones that are most satisfactory or useful to the users. The paper investigates the quality of RSs from a user-centric perspective. We discuss an empirical study that involved 210 users and considered seven RSs on the same dataset that use different baseline and state-of-the-art recommendation algorithms. We measured the user's perceived quality of each of them, focusing on accuracy and novelty of recommended items, and on overall users' satisfaction. We ranked the considered recommenders with respect to these attributes, and compared these results against measures of statistical quality of the considered algorithms as they have been assessed by past studies in the field using information retrieval and machine learning algorithms.

Keywords: Recommender systems, quality metrics, user study.

1 Introduction

Recommender Systems (RSs) play an increasingly important role in online applications characterized by a very large amount of data - e.g., multimedia catalogs of music, products, news, images, or movies. Their goal is to filter information and to recommend to users only the items that are likely of interest to them.

Traditionally, the quality of a RS is defined in terms of objective statistical metrics, e.g., error metrics and accuracy metrics, which do not involve users and are evaluated algorithmically, using well-known techniques developed in the fields of information retrieval and machine learning (e.g., hold-out or k-fold cross-validation).

More recently, RS research is exploring user-centric directions for measuring and improving the subjective quality of RSs. A number of researchers highlight the need of a shift of perspective, suggesting that the recommendations that are "best" according to objective metrics are sometimes not the ones that are most satisfactory or useful to the users [14]. Some works [14,15] pinpoint that the quality of the User eXperience (UX) with a RS as determined by its pragmatic factors (e.g., usability) or hedonic characteristics (e.g., aesthetics and "fun") are as important, or even more

P. Campos et al. (Eds.): INTERACT 2011, Part III, LNCS 6948, pp. 152–168, 2011.

important than algorithmically assessed quality to determine the user's attitudes towards a RS, and are more influential on users' decisions to use a system and to "purchase" recommended results.

User-centric approaches to RS quality evaluation have recently received some interest in the research and industry arena of RS and HCI communities. Still, empirical research in this area is currently in its early stage and a limited amount of user-based studies exist. Empirical research in this domain is rather costly, difficult in design and implementation, partially because of the intrinsic complexity induced by the high number of variables to be controlled, the computational sophistication of RSs, and the difficulty of involving need large datasets and a wide number of users.

The paper provides a contribution to this field discussing an empirical study that involved 210 users and considered 7 recommender systems, which use the *same* dataset and user interface, but implement 7 different baseline and state-of-the-art recommender algorithms. We measured the *user's perceived quality* of each RS, focusing on three attributes - perceived *accuracy*, *novelty*, and *global satisfaction*. We prioritize the considered recommenders with respect to these attributes and compare our results against the *objective statistical* quality of the considered algorithms as it has been assessed by past studies in the field, based on accuracy metrics. We discovered some interesting mismatches that suggest that objective metrics are not always good predictors of the perceived quality of RSs.

2 Background and Related Work

Recommender Systems (RSs) are generally classified into two families, characterized by different types of *recommender algorithms* [1]: *content-based filtering* (CBF) and *collaborative filtering* (CF).

In *CBF algorithms,* items are described by means of a set of explicit features. For instance, a movie can be characterized by genre, director, and list of actors. Such RSs tend to recommend items with the same characteristics as the movies a user "liked" in the past, thus they typically propose a limited variety of unexpected recommendations [9][23].

On the other hand, *CF algorithms* are based on collective preferences of the crowd: they recommend what similar customers bought or liked. Collaborative RSs are the most used, mainly because their implementation and integration in existing domains is relatively easy and their quality, in terms of objective metrics, is generally higher than CBF algorithms. However, some criticism is addressed also to CF recommenders, pinpointing that they are biased toward popularity, constraining the degree of diversity consumers would ever prefer [8].

Two families of *objective* metrics are typically adopted to automatically evaluate RSs: *error metrics* and *accuracy metrics* [9]. *Error metrics*, such as RMSE (Root Mean Square Error), measure the capability of the system to accurately estimate the ratings real users would give to item. *Accuracy metrics*, such as precision and recall, measure the effectiveness of the "top-N recommendation task" [7], i.e., the capability of a RS to *accurately select* a small set of items that the user will surely appreciate.

Both error and accuracy metrics can be automatically evaluated by means of well-known techniques developed in the field of machine learning, such as hold-out and leave-one-out (e.g., [7] and [22]). However, such standard metrics address a single property of RS's quality, relevance, while neglecting other issues that can be perceived as important by users but are more complex and articulated to operationalize.

As an alternative or complementary approach, a number of studies investigate a *user-centric* approach to RS evaluation, carrying on user-based empirical assessment or proposing conceptual frameworks for *perceived quality*.

Celma and Herrera [3] report an experiment exploring the users' perceived quality of novel recommendations provided by a CF and a CBF algorithm in the music recommendation context. Shearer [21] describes an experiment with 29 subjects on a movie RS to determine whether recommendations based on CF are perceived as superior to recommendations based on user population averages. The recommender systems suggested movies that subjects later viewed. Participants placed slightly more confidence in the CF recommendations with respect to the recommendations based on the population averages, but the perceived quality of the two algorithms was almost the same.

Ziegler et al. in [25] and Zhang in [24] propose diversity as a quality attribute: recommender algorithms should seek to provide optimal coverage of the entire range of user's interests. This work is an example of combined use of automatic and user-centric quality assessment techniques.

Pu and Chen [18] develop a framework called ResQue, which defines a wide set of user-centric quality metrics to evaluate the perceived qualities of RSs and to predict users' behavioural intentions as a result of these qualities. The framework provides 13 quality attributes and 60 questions that can be put to users for measuring them. Quality constructs are organized in four main classes: 1) "perceived *system* qualities", which refer to the functional and informational aspects of RRs (recommended items, interaction, and interface); 2) "beliefs" (user's perception on ease of use, usefulness, and control on interaction); 3) "attitudes" (the user's overall feeling towards a recommender, e.g., global user satisfaction, confidence and trust); 4) "behavioral intentions" (the degree at which a RS is able to influence users' decisions to implement the suggestions by the system). The framework represents an important contribution to understand the crucial factors that influence the user adoption of RSs, and provides a useful conceptual tool to guide the design and execution of user-centric evaluation studies of RSs. Several user-centric evaluations are reported in literature employing ResQue attributes [4,10,12,15,16,17]. They focus on different RSs (employing different user interfaces [15] or implementations, or datasets in different domains [16,17] (e.g., music [10] or film [12] or investigate the different perceptions of quality in culturally heterogeneous user groups [4], thus obtaining a variety of not comparable results.

As discussed in the next section, ResQue has been partially adopted also in our work. Still, our research differs from previous works in that it is more focused – it involves seven RSs which differ in terms of algorithms *only* - and also compares the results of perceived quality evaluation with objective quality measures of the considered algorithms.

3 Empirical Study of RS Perceived Quality

The general goal of the study was to compare measures of *user's perceived quality* against measures of *objective statistical quality of RSs* in order to provide some empirical evidence about the degree at which they are aligned, and to validate, or to confute, the hypothesis, underlying most existing studies, that objective statistical quality is a good predictor for user's perceived quality. The study was designed as a between subjects controlled experiment, in which we measured *perceived quality,* decomposed into a number of measurable attributes (*dependent variables*) in seven different experimental conditions, each one using a system that support the *same user interface,* employ the *same dataset* in the movie domain, but implements a *different* recommender *algorithm* (*independent variable*).

3.1 Perceived Quality Attributes

To better scoping our research, we focused our attention on three user-centric quality metrics:

Perceived accuracy (also called *Relevance*) - how much the recommendation matches the users' interests, preferences and tastes;
Novelty - the extent to which users receive "new" recommended items;
Overall users' satisfaction - the global users' feeling of the experience with the RS.

Considering the classification of the ResQue model [18], the third metric belongs to the category "Attitudes", while the first two attributes fall in the category "Perceived System Qualities", and, in particular, the subcategory "Quality of recommended items".

Our notion of perceived accuracy is meant in the same way as in the ResQue model. Still, we operationalize their measure is a slightly different way w.r.t. ResQue, as discuss in the following section.

Our concept of novelty can be regarded as a sub-dimension of ResQue novelty, which encompasses not only the idea of "new" but also of "interesting" and "surprising", the latter being referred to as "serendipity" in Herlock [9]. In addition, we distinguish between two "levels" of novelty, called respectively "First Order Novelty (FON)" and "Second Order Novelty (SON)". FON is a weaker for of novelty: it considers a movie to be *novel* for a user only if he/she has never watched it (without discriminating whether he/she has any knowledge about it). SON is more stringent concept and subsumes FON: a recommended movie is considered novel if the user has *no* idea of it. SON is a more conservative way to measure novelty, and, as we will see in the next sections, leads to lower values than FON. Anyway, it is interesting to compute the novelty in both ways and to compare the obtained results.

3.2 Algorithms

Our study considered several state-of-the-art recommender algorithms: (i) one non-personalized algorithm used as baseline, referred to as TopPop, (ii) five *collaborative* algorithms - CorNgbr, NNCosNgbr, AsySVD, and two versions of PureSVD - and (iii) a *content-based* one – LSA. In the following we provide a short description of each algorithm. Further details can be found in [5] and in the papers quoted therein.

3.2.1 Non-personalized Algorithm

TopPop (Top Popular) implements a simple, non-personalized estimation rule, which recommends the most popular items to any user, regardless his or her ratings. Such algorithm serves as baseline for the more advanced personalized algorithms.

3.2.2 Collaborative Algorithms

There are two major approaches to collaborative filtering: (i) the neighborhood approach and (ii) the latent factor approach.

Neighborhood models

Neighborhood models represent the most common approach. Rating prediction is based on the similarity relationships among either users or items, in terms of collected ratings. Item-based similarity is usually preferred to user-based similarity for its better performance in terms of RMSE and its higher scalability [20]. Prior to computing similarities, it is advised to remove a set of biases in the collected ratings, such as: (i) *user effects*, which represent the tendency of some users to rate higher than others, and (ii) *item effects*, which represent the tendency of some items to be rated higher than others. Typically, only the most similar items – referred to as neighbors - are taken into consideration. In our experiments, the neighborhood size has been set to 200.

CorNgbr (Correlation Neighborhood) is a classical technique that computes item-item similarity by means of the Pearson linear correlation coefficient [13].

Similarly, *NNCosNgbr* (Non-normalized Cosine Neighborhood) computes item-item similarity by means of the cosine coefficient. Unlike Pearson correlation - which is computed only on ratings shared by common rater - the cosine coefficient is computed over all ratings, taking missing values as zeroes. In addition, while CorNgbr averages the ratings received by similar items, NNCosNgbr simply sums up such ratings, higher ranking items with more similar neighbors [5].

Latent factor models

Latent factor models - also informally known as SVD models after the related Singular Value Decomposition (SVD) - represent users and items as vectors in a common low-dimensional 'latent factor' space. In such a space, users and items are directly comparable and the rating of a user u on an item can be estimated as the proximity (e.g., inner-product) between the related latent factor vectors. This family of algorithms has been leading the Netflix contest thanks to its performance in terms on RMSE.

AsySVD (Asymmetric SVD) is a powerful matrix factorization model that reported an RMSE of 0.9000 in the Netflix context. Differently from other latent factor models, AsySVD represents users as a combination of item features. Thus, AsySVD is able to immediately compute recommendations for users not yet parameterized and to adjust recommendations as fast as the user being recommended enters new ratings, providing an immediate feedback to his or her activity [13].

PureSVD is a latent factor algorithm recently proposed [5], whose rating estimation rule is based on the conventional SVD. In order to use conventional SVD – which is not defined for matrices with missing values – unknown ratings have been treated as

zeros. We have tested PureSVD with two different sizes of latent factors: 50 and 300. In fact, the larger the number of latent factors the more the algorithm is able to detect the uniqueness in users' taste. The smaller the number of latent factor the more the algorithm tends to recommend the most popular items.

3.2.3 Content-Based Algorithms

Content-based algorithms recommend items whose content is similar to the content of items the user has positively rated in the past. For instance, in the domain of movies, such content can be the movie title, the playing actors, the director, the genre, and the summary. While the basic approach to content-based recommendations is based on the analysis of term-by-item occurrences, and neglecting the semantic structure of item content, more advanced techniques try to exploit such semantic features. In our experiments we have used *LSA* (Latent Semantic Analysis), a well-known method in the field of information retrieval for automatic indexing and searching of documents. The approach takes advantage of the implicit structure (i.e., latent semantic) in the association of terms with documents. Such semantic structure comes out by representing the term-by-item relationships in a low-dimensional 'latent factor' space computed through SVD [11].

3.3 Instruments

3.3.1 Technological Framework

To run our experiments, we used a web-based commercial recommender framework - called ContentWise (www.contentwise.tv - Figure 1). ContentWise supports users with a wide range of typical RS functionalities, such as browsing a catalog of products, retrieving the detailed description of each item, rating it, getting recommendations and rating their relevance. The modularization and customization features of the system allowed us to easily create different experimental conditions by implementing different algorithms while maintaining interface and dataset invariant.

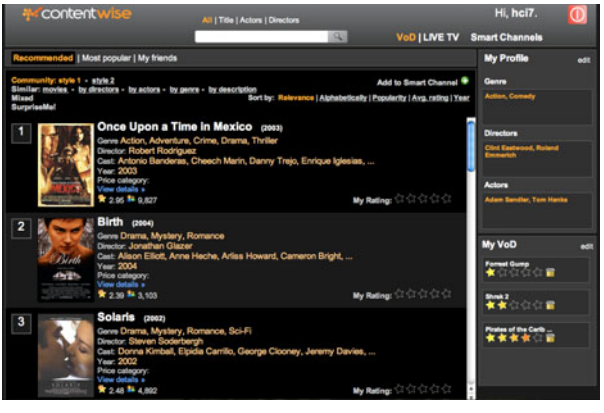
3.3.2 Dataset

The dataset included 2137 movies and about 7.7 million ratings given by 49,969 users. The data consider a subset of the well-known large-scale movie dataset Netflix, published for the purpose of the famous contest organized by the homonymous movie rental American provider. In addition, for the purpose of our study, these contents were integrated with data and metadata (e.g., movie plot, images, actors, director and genre) collected online.

3.3.3 Data Collection Technique

As discussed more precisely in the following section, the chosen user-centric metrics were measured using a *questionnaire* that evaluators completed for each user during the experiment. It collects both users' demographic attributes and their opinions about *perceived accuracy*, *novelty* and *overall satisfaction.*

(a)



(b)

(c)

Fig. 1. ContentWise interface: a) initial exploring and rating of movies (up); b) results of recommendations; c) movie details

3.4 Participants

Data collection was carried on by a team of 14 master students (two per experimental condition) selected among the best ones of those attending two courses - HCI and iTV - at our School of Information Engineering. Students were motivated in performing the evaluation to the best of their capabilities, for a number of reasons. This work represented the second assignment proposed at our courses, and accounted for 50% of the final mark. Students were pre-screened as they had to pass brilliantly the first part of our exam in order to be eligible for performing this one. In addition, they freely selected this assignment from a set of others proposed by teachers.

Students were initially trained by us to perform the study, were given written instructions on the evaluation procedure, and were regularly supervised by a teaching assistant during their activities. After a pre-screening among school maters, friends and relatives, each pair of student evaluators recruited a group of *thirty* subjects for each algorithm, almost uniformly distributed w.r.t. to gender and age. Overall, the study involved *210* users aged between 20 and 50; 54% subjects were male and 46% female. None of them had been previously exposed to the system used in our study nor had technical knowledge about RSs.

3.5 Procedure

The evaluation took place in informal environments such as university (15%), interviewer's place (32%), and interviewee's place (31%). Each interview lasted from 15 to 35 minutes. The motivation for such a temporal variability is that in case of completely novel recommendations, users were invited to explore information related to unknown items (see below) in order to express more precise and conscious opinions on the quality of the RS used.

Each participant was initially asked to provide his/her personal information (age, gender, education, nationality, and number of movies watched per month). Afterward, (s)he was invited to browse the movie catalog using the ContentWise system (pre-customized on a specific algorithm). The user was then asked to freely select five known (not necessarily watched) movies and rate his/her degree of appreciation or interest for them using a 1-5 point scale (1 = low interest for/appreciation of the movie; 5 = high). On the basis of these ratings, five recommendations were returned by the system (using the current algorithm). The user was finally invited to explore the results and reply to a set of questions related to the quality of the recommendations.

Novelty measures were collected as follows. For each recommended item we first asked the question "Have you ever watched this movie?" This answer (yes/no) was used to compute First Order Novelty (*FON*). FON for an item is 1 if the user has *never* watched the movie and 0 otherwise. If the user has *never* watched a recommended movie (FON=0), we proceeded with an in depth exploration to assess Second Order Novelty (SON). We asked the user if (s)he had ever *heard* about the movie, inviting him/her to explore the information related to the movie (director, cast, abstract, trailer,...) to refresh her memory. If a user answers "yes" to the above question (or if FON is 0), SON is set to 0, while it is set to 1 otherwise.

Perceived accuracy measures were collected as follows. For each recommended movie, if the user had already watched it, he/she was asked to rate how much he/she

liked/disliked (on a 1-5 scale). Otherwise, if the user had already seen the trailer, he/she was invited to rate the degree of potential interest for the movie. If the user had never been exposed to the movie or its trailer, he/she was invited to look at the trailer and to explore additional information (e.g., director, the actors, and so forth) and then to give a rating of potential interest. For each user, each of the above attributes – FON, SON and perceived accuracy is calculated as the average on the respective values assigned to each recommended item.

Finally, the overall satisfaction was computed by asking each subject to provide a global judgment (in a 1-5 rating scale) about the list of recommended movies and was allowed to express a free comment.

4 Empirical Study Results

In this section we present the user-centric metrics of relevance and novelty computed on the basis of the questionnaire data.

4.1 Accuracy

Users' perception of the accuracy of the recommendations is measured by considering, for each user and each suggested movie, the user's opinion on the movie expressed in a 1-5 scale. Figure 2 shows the box plot of the perceived relevance for each algorithm. Upper and lower ends of boxes represent 75th and 25th percentiles. Whiskers extend to the most extreme data point which is no more than 1.5 times the interquartile range. Median is depicted with a solid line, mean with a dot. Outliers are represented with empty circles.

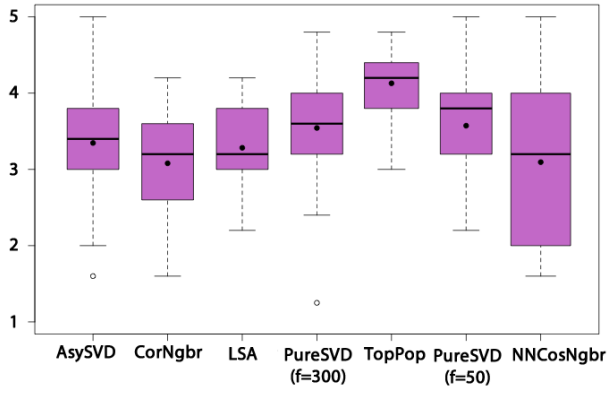


Fig. 2. Perceived relevance for each algorithm. Relevance ranges from 5 (most relevant) to 1 (not relevant).

We can see that all the algorithms have an average relevance between 3 and 4. This result shows that, on average, users are satisfied by the quality of the recommendations (the median for all the algorithms is greater than 3). Moreover, 75%

of the users have received relevant recommendations from five of the considered algorithms (AsySVD, LSA, PureSVD50 PureSVD300 and TopPop). Only NNCosNgbr produces a relatively large number of bad recommendations (25% of the recommendations are rated 2 or less).

The most surprising result is the TopPop algorithm, having the largest perceived accuracy. This result is surprising because the TopPop algorithm suggests to all users the same list of 5 movies, without taking into consideration the user profile. These movies are: "Pirates of the Caribbean: The Curse of the Black Pearl"; "Forrest Gump"; "The Lord of the Rings: The Two Towers"; "The Lord of the Rings: The Fellowship of the Ring"; "The Sixth Sense".

According to our study, any user found in this list an average of four interesting movies (more than 80% of the users rated TopPop recommendations with 3 or more stars). This result may provide evidence against the real usefulness of sophisticated recommender algorithms, a hypothesis that will be further analyzed in the following paragraphs and in the discussion section.

4.2 Novelty

A similar analysis was performed for perceived novelty. Novelty refers to the previous knowledge of the user about the suggested movies. Unlike relevance, novelty measures are based on two questions, respectively responded with either "yes" (novelty value = 1= totally novel recommendations) or "no" (novelty value = 0= no novel recommendations). Figure 3 shows the box plot of perceived first-order novelty (percentage of never-watched movies in the recommendation list). Similarly, Figure 3 shows the second-order novelty (percentage of never-heard-of movies in the recommendation list). By definition, first-order novelty (FON) is always greater than or equal to second-order novelty (SON). The two metrics provide quite similar results and are strongly correlated (the correlation factor is 0.8).

If perceived relevance was on average satisfactory across all the algorithms, the same cannot be said for novelty. On average, no algorithm was able to suggest more

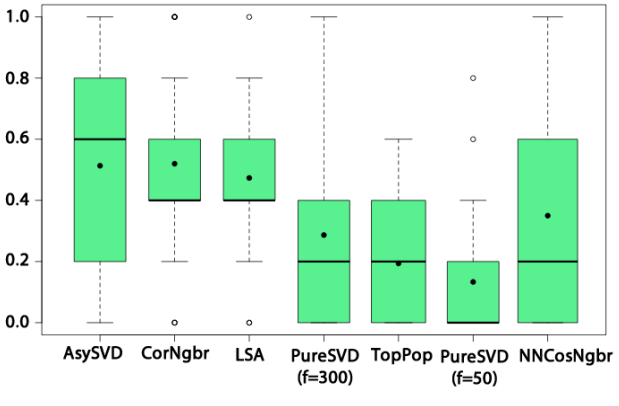


Fig. 3. Perceived First-Order Novelty (never watched) for each algorithm

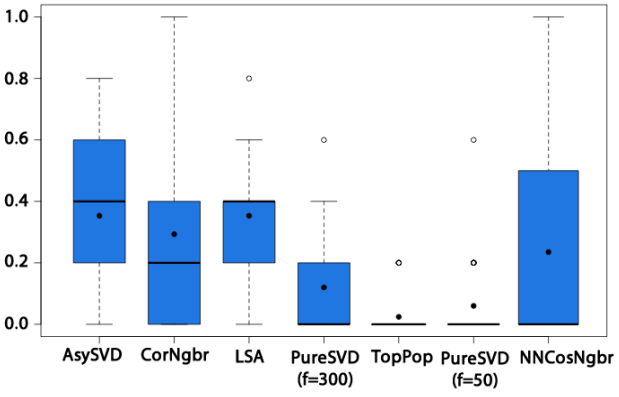


Fig. 4. Perceived Second-Order novelty (never heard of) for each algorithm

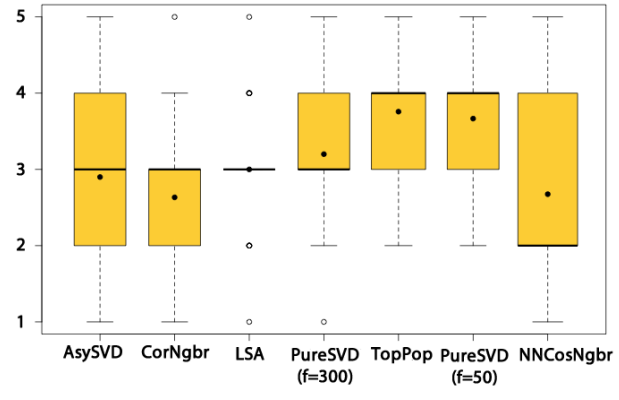


Fig. 5. Global Satisfaction for each algorithm. Satisfaction ranges from 5 (totally satisfied) down to 1 (no satisfied at all).

than 40% of totally-unknown-to-the-user movies (SON). Moreover, TopPop and PureSVD50 were not able to suggest novel movies at all (with the exceptions of few outliers, SON is always 0%).

Finally, Figure 5 shows the results for global user satisfaction. User satisfaction was measured according to a 1-5 points scale. Contrary to the relevance, the collected responses have a large variance and there seems to be no agreement in users' opinion, at least with AsySVD, CorNgbr, LSA and PureSVD300 (median and average equal or close to 3). In order to better compare the results, we first used 1-way ANOVA. The test suggests that, for each of the dependent variables, at least one of the algorithms differs significantly with respect to the others. We run multiple pair-wise comparison post-hoc tests using Tukey's method. All tests were run using a significance level $\alpha = 0.05$. Although no algorithm is significantly better (or worse) than all the other in terms of any of the quality dimensions, we can at least identify a partial order, as outlined in Table 1.

Table 1. Partial Ordering of RSs w.r.t. the various quality attributes

	Accuracy	**Novelty**	**Global satisfaction**
Maximal	TopPop	AsySVD LSA CorNgbr	PureSVD50 TopPop
Intermediate	AsySvd PureSVD300 PureSVD50	NNCosNgbr PureSVD300	PureSVD300 LSA
Minimal	NNCosNgbr LSA CorNgbr	PureSVD50 TopPop	AsySVD CorNgbr NNCosNgbr

According to this ordering, TopPop is the maximal algorithm in term of relevance (i.e., the algorithm with the best perceived relevance), while NNCosNgbr, CorNgbr and LSA are the minimal algorithms (i.e., the algorithms with the worst perceived relevance).

We have performed the same comparison for first-order and second-order novelty. Being the two novelty metrics correlated, the comparisons define the same partial ordering. According to this ordering, AsySVD, CorNgbr and LSA are the algorithms with the best perceived novelty, while TopPop and PureSVD50 are the algorithms with the worst perceived novelty.

The last column of the table shows the partial ordering according to the global satisfaction; TopPop and PureSVD300 are the algorithms which mostly satisfied the users, while AsySVD, CorNgbr and NNCosNgbr are the algorithms which less satisfied the users.

5 Objective Evaluation of Quality

5.1 Objective Metrics and Their Evaluation Method

As mentioned in section 2, RS performance is traditionally usually measured using objective metrics. In particular, there are methodologies based on accuracy metrics (e.g., precision, recall and fallout) and on error metrics (e.g., RMSE and MAE). Some of the algorithms tested in this study (TopPop, NNCosNgbr and PureSVD) cannot be evaluated with error metrics [9]. Hence, we considered only accuracy metrics in our study. In particular, we focused our attention on recall r (the conditional probability of suggesting a movie given it is relevant for the user) and on fallout f (the conditional probability of suggesting a movie given it is irrelevant for the user).

A good algorithm should have high recall (i.e., it should be able to recommend items of interest to the user) and low fall-out (i.e., it should avoid to recommend items of no interest to the user).

A measure that combines recall and fall-out is the F-measure. F-measure is defined as the harmonic mean of precision and recall. Precision can be estimated from recall and fall-out by using the definition provided in [19].

The testing methodology adopted in this study is similar to the one described in [5]. The known ratings of the dataset are split into two subsets: training set *M* and test set *T*. The test set *T* contains only 5-stars ratings. Therefore we can reasonably state that *T* contains items relevant to the respective users. The detailed procedure used to create *M* and *T* from the Netflix dataset is similar to the one used for the Netflix prize, maintaining compatibility with results published in other research papers [2,5].

In this work, the training set *M* is a subset of the original Netflix training set, while the test set *T* contains only part of the 5-stars ratings from the Netflix probe-set. The test set contains 69,039 5-star ratings.

In order to measure recall, we first trained the algorithm over the ratings in *M*. Then, for each item *i* rated 5-stars by a user *u* in *T*, we followed these steps:

1. We randomly selected 1,000 additional items unrated by user *u,* assuming that the user *u* is not interested in most of them.
2. We predicted the ratings for the test item *i* and for the additional 1,000 items.
3. We formed a top-5 recommendation list by picking the 5 items with the largest predicted ratings.

The overall recall *r* was computed as

$$r = \frac{\text{\# times the element is in the list}}{\text{\# elements in } T}$$

A similar approach was used to measure fall-out, with the only difference being in the composition of the test set *T*, that now contains only 1-stars ratings. The fall-out *f* is computed as

$$f = \frac{\text{\# times the element is in the list}}{\text{\# elements in } T}$$

5.2 Objective Metrics Evaluation Results

Table 2 presents the objective accuracy of the tested algorithm. Algorithms in the table are ordered in decreasing order of recall. Recall and F-measure suggest

Table 2. Recall, fallout and F-measure computed for Top-5 recommendation lists

	Type	Recall	Fallout	F-measure
PureSVD50	Collaborative Latent factors	0.29	0.005	0.45
PureSVD300	Collaborative Latent factors	0.25	0.005	0.40
AsySVD	Collaborative Latent factors	0.13	0.001	0.23
NNCosNgbr	Collaborative Item-based	0.12	0.010	0.21
TopPop	Collaborative Non-personalized	0.11	0.025	0.20
CorNgbr	Collaborative Item-based	0.08	0.010	0.15
LSA	Content	0.01	0.002	0.02

PureSVD as being the most accurate algorithm. Second in line are AsySVD, the two item-based neighborhood algorithms and the non-personalized TopPop algorithms, all of them with a similar recall. The content-based LSA algorithm has the worst accuracy both in terms of recall and F-measure. If we look at fallout, AsySVD and LSA obtain the best results, while NNCosNgbr and TopPop are the algorithms with the largest error rate.

6 Discussion

The analysis of the results presented in the previous sections suggests a number of interesting considerations: 1) simple, non-personalized algorithms are well perceived by the users; 2) the perceived novelty of content-based recommendations is equal or even better with respect to collaborative recommendations; 3) objective accuracy metrics (e.g., recall and fallout) are not a good approximation of user perceived quality.

Let's start from the first point. According to Figure 2, no algorithm is significantly better (or worse) than all the others in terms of *perceived relevance*. However, the partial ordering among the algorithms (Table 1) highlights that *TopPop* is the algorithm with the best perceived relevance (this is unexpected) and with the worst novelty (as expected), thus its utility is limited because oftentimes the user has already watched the suggested items. Still, *TopPop* (together with *PureSVD300*) is at the top level in terms of global user satisfaction. In summary: *simple non-personalized TopPop recommendations are better perceived by the users with respect to other more sophisticated and personalized recommender algorithms,* although users are aware of the low utility of such recommendations. Global user satisfaction seems mainly driven by the perceived accuracy than by the novelty of the recommendations. This is a somehow surprising result, especially if we consider the large academic and industrial effort in the development of new and more sophisticated recommender algorithms.

As for novelty, Table 2 highlights that AsySVD, CorNgbr and LSA are the algorithms with the best perceived novelty, while TopPop and PureSVD50 are the algorithms with the worst perceived novelty. Thus, the perceived novelty of content-based recommendations is equal or even better with respect to collaborative recommendations. *This result is in contrast with most of the existing literature in RS, which considers content-based algorithms as not able to recommend novel items* (see, e.g., [9] and [23]). To try an interpretation of this result, we should consider that collaborative algorithms, by design, are biased toward popular "Blockbuster" items, thus reducing the chances of novel recommendations. Collaborative algorithms are trained (e.g., tuned) to achieve the best performance in terms of objective accuracy. Because objective accuracy is computed on already-rated items, collaborative algorithms cannot recommend items with limited historical data. This creates the rich-get-richer effect for popular items and the opposite effect for unpopular ones, which results in lower novelty. As a consequence, collaborative algorithms tend to reinforce the popularity of already popular items and to recommend mainly common movies, which are likely not to be novel.

Finally, the comparison between Tables 1 and 2 shows the lack of correspondence between *objective accuracy metrics (e.g., recall and fallout) and users' perceived quality*. In other words, objective quality attributes are not good predictors of users' perceived quality of a recommender algorithm, at least in our case. To try an interpretation of this phenomenon, it is useful to consider that objective metrics compute accuracy of recommendations by (i) exploiting previously rated movies, i.e., user's rankings of movies that they know about, and (ii) sampling all the ratings in the dataset - the majority of which concern few popular movies. Consequently, objective metrics focus their attentions on measuring the quality of an algorithm when recommending popular items and might not be particularly effective for measuring the quality of the same algorithm when recommending novel, unrated items.

7 Conclusions

In this work we have investigated under different perspectives the quality of 7 RSs that only differ in terms of recommender algorithms. We first measured quality from a user-centric perspective and then compared these results against measures of statistical quality, in terms of recall and fallout. The considered RSs include both state-of-the-art techniques and a trivial non-personalized recommender algorithm. There are three main interesting findings:

(i) the simple, non-personalized algorithm is well perceived in terms of overall user satisfaction, although users are aware of the low utility of such recommendations;
(ii) the perceived novelty of content-based recommendations is equal or even better with respect to collaborative recommendations;
(iii) statistical accuracy metrics (e.g., recall and fallout) are not necessarily a good approximation of the quality perceived by the users.

Our research has its limitations. First, the sample size of participants used for each RS (30) is relatively small. Still, the fact that we replicated the study in seven experimental conditions using the same methodological framework, and involving overall 210 tested subjects, partially compensates for this drawback and strengthens the reliability of our results. Second, we focused our investigation of user perceived quality on a small set of attributes – perceived accuracy, novelty, and overall user satisfaction. Other approaches, e.g., the ResQue model, include many additional user-centric metrics, which we did not consider in our study. Our choice may be regarded as a weakness, but it was motivated by the need to keep data collection workload affordable. ResQue provides a 60 items questionnaire. Administrating so many questions could have been too demanding for respondents and too time-consuming for data collectors, considering our goal of collecting measures in 7 experimental conditions. In addition, we sought to focus on those user-centric attributes that are more related to standard objective quality metrics and thus are more comparable with them. Objective metrics are related to the quality of recommend items, thus we gave higher priority to measuring ResQue attributes related to these aspects. Certainly, other measures of user perceived quality are worth being investigated in relationship to objective quality, and we are planning to replicate our study in order to include them.

In spite of the above limitations, our work provides contributions both from a research and practical perspective. To our knowledge, this is the first work that systematically compares perceived quality in a significant number of different RSs isolating a precise factor – the underlying recommender algorithm – and analyzing the results against statistical, objective measures of quality. For the practice of RS design and evaluation, our results may promote further approaches that move beyond the attention to conventional accuracy metrics and shift the emphasis to more user-centric factors.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. on Knowledge and Data Engineering 17(6), 734–749 (2005)
2. Bennett, J., Lanning, S.: The Netix Prize. In: Proceedings of KDD Cup and Workshop, pp. 3–6 (2007)
3. Celma, Ò., Herrera, P.: A new approach to evaluating novel recommendations. In: RecSys 2008: Proc. of the 2008 ACM Conf. on Recommender Systems, pp. 179–186. ACM, New York (2008)
4. Chen, L., Pu, P.: A cross-cultural user evaluation of product recommender interfaces. In: Proc. of the 2008 ACM Conf. on Recommender Systems, RecSys 2008, pp. 75–82. ACM, New York (2008)
5. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-N recommendation tasks. In: RecSys 2010: Proc. of the Fourth ACM Conf. on Recommender Systems, pp. 39–46. ACM, Barcelona (2010)
6. Cremonesi, P., Turrin, R.: Analysis of cold-start recommendations in IPTV systems. In: RecSys 2009: Proc. ACM Conf. on Recommender Systems, pp. 233–236. ACM, New York (2009)
7. Deshpande, M., Karypis, G.: Item-based top-N recommendation algorithms. ACM Trans. Information Systems (TOIS) 22(1), 143–177 (2004)
8. Fleder, D., Hosanagar, K.: Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. Management Science 55(5), 697–712 (2009)
9. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. on Information Systems (TOIS) 22(1), 5–53 (2004)
10. Hu, R., Pu, P.: Acceptance issues of personality-based recommender systems. In: Proc. of the Third ACM Conf. on Recommender Systems, pp. 221–224. ACM, New York City (2009)
11. Husbands, P., Simon, H., Ding, C.H.Q.: On the use of the singular value decomposition for text retrieval. Computational Information Retrieval, 145–156 (2001)
12. Jones, N., Pu, P.: User Technology Adoption Issues in Recommender Systems. In: Proc. of the 2007 Networking and Electronic Commerce Research Conf., Riva del Garda, Italy, pp. 379–394 (2007)
13. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: KDD 2008: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434. ACM, New York (2008)
14. McNee, S.M., Riedl, J., Konstan, J.A.: Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: CHI 2006 Extended Abstracts on Human Factors in Computing Systems, pp. 1097–1101. ACM, New York City (2006)

15. Pu, P., Chen, L.: Trust building with explanation interfaces. In: Proc. of the 11th int. Conf. on Intelligent User Interfaces, IUI 2006, pp. 93–100. ACM, New York (2006)
16. Pu, P., Chen, L., Kumar, P.: Evaluating product search and recommender systems for e-commerce environments. Electric Commerce Research Journal 8(1-2), 27 (2008)
17. Pu, P., Zhou, M., Castagnos, S.: Critiquing recommenders for public taste products. In: Proc. of the Third ACM Conf. on Recommender Systems, RecSys 2009, pp. 249–252. ACM, New York (2009)
18. Pu, P., Chen, L.: A User-Centric Evaluation Framework of Recommender Systems. In: Proc. of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain (September 2010)
19. Raghavan, V., Bollmann, P., Jung, G.S.: A critical investigation of recall and precision as measures of retrieval system performance. ACM Trans. Inf. Syst. 7, 205–229 (1989)
20. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-based collaborative filtering recommendation algorithms. In: 10th Int. Conf. on World Wide Web, pp. 285–295 (2001)
21. Shearer, A.W.: User response to two algorithms as a test of collaborative filtering. In: CHI 2001 Extended Abstracts on Human Factors in Computing Systems, pp. 451–452. ACM, New York (2001)
22. Takács, G., Pilászy, I., Németh, B., Tikk, D.: Scalable collaborative filtering approaches for large recommender systems. The J. of Machine Learning Research 10, 623–656 (2009)
23. Weng, L., Xu, Y., Li, Y., Nayak, R.: Improving recommendation novelty based on topic taxonomy. In: Proc. of the 2007 IEEE/WIC/ACM International Conf. on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IATW 2007, pp. 115–118. IEEE Computer Society, Washington, DC, USA (2007)
24. Zhang, Y., Callan, J., Minka, T.: Novelty and redundancy detection in adaptive filtering. In: Proc. of the 25nd ACM SIGIR Conf. on R&D in Information Retrieval, pp. 81–88. ACM, New York City (2002)
25. Ziegler, C.N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: Proc. of the 14th International Conf. on World Wide Web, pp. 22–32. ACM, New York (2005)