2012 Sixth International Conference on Complex, Intelligent, and Software Intensive Systems

User-centered Evaluation of Recommender Systems with Comparison between Short and Long Profile

Francesco Epifania
Dipartimento di Scienze dell'Informazione (DSI)
Università degli Studi di Milano
Milan, Italy
francesco.epifania@dsi.unimi.it

Paolo Cremonesi
Dipartimento di Elettronica e Informazione (DEI)
Politecnico di Milano
Milan, Italy
cremonesi@elet.polimi.it

***Abstract*—The growth of the social web poses new challenges and opportunities for recommender systems. The goal of Recommender Systems (RSs) is to filter information from a large data set and to recommend to users only the items that are most likely to interest and/or appeal to them. The quality of a RS is typically defined in terms of different attributes, the principal ones being relevance, novelty, serendipity and global satisfaction. Most existing works evaluate quality of Recommender Systems in terms of statistical factors that are algorithmically measured. This paper aims to explore whether (i) algorithmic measures of RS quality are in accordance with user-based measure and (ii) the user perceived quality of a RS is affected by the number of movies rated by the user. For this purpose we designed, developed and tested a web recommender system, TheBestMovie4You (http://www.moviers.it), which allows us to collect questionnaires about the quality of recommendations. We made a questionnaire and gave it to 240 subjects and we wanted to have as wide a set of users as possible using social web. In a experiment we asked the users to choose five movies (short profile), in a second to choose ten (long profile). Our results show that statistical properties fail in fully describing the quality of algorithms, because with user-centered metrics we can outline an algorithm's features that otherwise could not be detected. The comparison between the two phases highlighted a difference only in three cases out of twenty.**

***Index Terms*—Recommender Systems, information filtering, algorithmic evaluation, user-centered evaluation, short profile, long profile**

I. Introduction

The social web has turned information consumers into active contributors providing large amount of ratings about the consumed information items. In this context, finding relevant and interesting content is challenging for existing recommender approaches. RS technology plays an important role in web-based applications characterized by a very large amount of data. Recommendations are generated on the basis of different elements, e.g., popularity, demographic information, the user's past preferences or choices, the user's explicit ratings on a sample of proposed items. Most existing works in the domain of RS evaluation do not involve users and operationalize quality in terms of statistical properties that are evaluated algorithmically (i.e., automatically), called objective metrics. The effectiveness of a RS is clearly related to the quality of its recommendations. RS quality is defined in terms of different attributes [1], the most important being Relevance, Novelty,and Global Satisfaction [2], [3] and to which we added Serendipity. Relevance refers to the ability of a RS to provide items that fit the user's preferences. Novelty measures what is new to the user about the recommendation process. We divided this latter metric in two orders, called first order novelty (FON) and second order novelty (SON). Serendipity gives a value to the surprise factor, while Global Satisfaction mirrors the response to the "Rate your recommendation experience" element in the questionnaire [4], [5].The above definition of quality is intrinsically "user-centered": it formulates quality in terms of what is perceived as valuable by the end users. Our research i) expands [5] exploring whether automatic measures of RS quality are in accordance with user-based quality measures and ii) explores whether the quantity of items rated by the users affects recommendation quality through a two-parts experiment (namely, "short profile" and "long profile") that involved 240 users and considered four RSs which share the same dataset and user interface. The participants in this data collection were colleagues, students, friends and family members and we wanted to have as wide a set of users as possible using social web, social network and social space. We helped users on-site or using Skype or Facebook's chat. In this paper we present a study that considers different recommendation algorithms (collaborative, content-based and non-personalized filtering) in the movie domain and then evaluates them using both algorithmic and user-centered quality assessment techniques. In order to expand [5], we used 4 algorithms : we chose DirectContentKNN (a recently developed algorithm) as content-based algorithm and MovieAVG as non-personalized algorithm, these algorithms are different than the previous experiment, PureSVD and AsySVD as collaborative algorithms. Here we analyze the data resulting from our study and discuss possible implications for RS research and practice.

II. Related Works

Celma and Herrera in [6] described an experiment which studied the users' perceived quality of recommendations provided in the music recommendation context. Shearer in [7] describes an experiment to determine whether recommendations based on collaborative filtering are perceived as superior to recommendations based on non-personalized average ratings, with the result of a slightly superior confidence in collaborative

978-0-7695-4687-2/12 $26.00 © 2012 IEEE
DOI 10.1109/CISIS.2012.114

algorithms. Pu and Chen in [8] developed and described a framework called ResQue. It defines a wide set of user-centric quality metrics to evaluate the perceived qualities of RSs and to predict users' behavioral intentions as a result of these qualities. The framework defines thirteen quality attributes extracted from sixty questions asked to the user after the recommendation. They are divided into four classes ("perceived system qualities", "beliefs", "attitudes", "behavioral intentions"). In the literature ResQUE [9]–[14] has become a model for the analysis of user-centered qualities. In our work, too, we adopt it partially. We decided to reduce the length of the questionnaires, following the choice of [5]focusing only on some of the parameters described in the model. Unlike similar research, our work isolates the recommender instrument and focuses on the differences between algorithms; we also compare the results of perceived quality evaluation with objective quality measures of the considered algorithms.

III. Algorithm Description

In this section we describe our dataset and the algorithms used in our study.

A. Dataset

1) User Rating Matrix (URM): The URM contains the ratings of a set of movies from a large amount of users. Our URM is a subset of the data taken from the Netflix [1] challenge, the famous American online rental service, and contains about 6,500 items and about 248,000 users, with a density of about 0.5% (about 9,000,000 ratings). It's an improvement from the dataset used in the previous experiment [5], which contained 2,137 items and 49,969 users, with a density of about 7%. Naturally, all redundancies were removed (Netflix stores different versions of the same movie) from the URM. This dataset is the same for every algorithm, and in this version it never gets updated. This matrix is mainly used by collaborative and non-personalized algorithms.

2) Item Content Matrix (ICM): The ICM is a matrix that describes the relations between items and metadata. Each matrix element contains a parameter that indicates how much an item is correlated to the selected metadata. The metadata include actors, directors, genres, and plot elements. This matrix is mainly used by content-based algorithms.

B. Algorithm Families

Recommender algorithms are usually grouped into three families, according to the way they compute the results. In our study we focused on the behavior of four algorithms, PureSVD and AsySVD are collaborative algorithms, DirectContentKNN is content-based and MovieAVG is non-personalized.

1) Collaborative Filtering Algorithms: Collaborative algorithms compute the recommendation looking for similarities between the user who asks for recommendation and other system users (in our case, taken from the User Rating Matrix).

These algorithms do not use the content of the items and focuses on exploit "collective preferences of the crowd" [15], [16]. They use rating profiles as source data and suggest items voted positively from users with similar tastes. This kind of technique is one of the most used in RS, because it's relatively easy to implement and has often a positive response from the users. The problem is that this kind of filtering tends to suggest popular movies and fails to reach items in the long tail (i.e., the degree of diversity of the items is pretty small). The direct consequence therein is that items with no rating will never be recommended by these algorithms. We chose to use AsySVD and PureSVD [17] .

2) Content Based Filtering Algorithms: CBF algorithms recommend items finding similar items in the catalog. For example, if one user gave positive rates to some Steven Spielberg movies, the system will suggest movies directed by him for which the user has not yet expressed a vote. The data used is this analysis are the actors, the directors, the genre, and plot elements. We chose to use DirectContent KNN. Direct Content is a method based on vectorial representation of the items which, differently from Cosine Content doesn't normalize data. This means that calculated similarities contained in the model will be different from Cosine Content. The model is generated as the product between URM and its transpose in order to obtain a similarity matrix (SM) with dimensions item * item. Recommendation list (rec_list) will instead be generated ad the product between the user profile and SM matrix.

$$model = URM \cdot URM^T$$

$$rec\_list = user\_profile \cdot URM$$

This algorithm uses k-NN filtrage. This operations considers only the k items similar to the one which is recommended and ignores the others.

3) Non Personalized Filtering Algorithms: Non-personalized filtering algorithms are the simplest family. Their recommendations are completely independent of the user's choices. Although it may seem strange, previous research indicates that according to some of the metrics sometimes non-personalized algorithms get a better reception from the users than CBF and CF algorithms. We chose to use MovieAvg that calculates the average rate for each item, orders them in an descending way and returns the five movies with the highest average rate. Results may vary according to each dataset because, for example, if a movie is linked to a few rates but with a high value, acquires priority in the list more than a movie with lots of ratings but with a lower value. In our case the suggested movies were: 1. Anne Of Green Gables 2. The Incredibles 3. Band Of Brothers 4. Inuyasha 5. The Lord of the Rings: The Return of the King

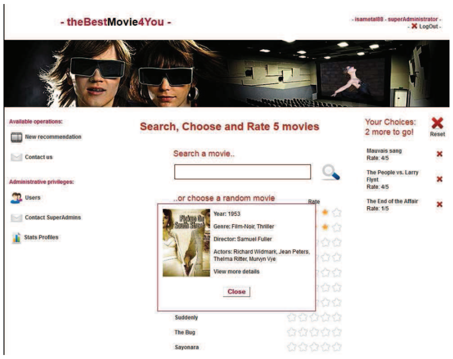
IV. Web System Description

Every user who wants to use the website has to register. S/he is asked for a nickname, a password (in order to log in at a later date), an email address (in order to write to the user if necessary) and some personal data: date of birth, gender,

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

level of education, average number of movies seen per month, nationality.

- The system automatically activates the user, assigns an algorithm following our needs, and establishes his/her powers and permissions. All administrators receive an e-mail notification of the registration.
- After registration the user is asked to choose and rate five movies from our database. We present a list of ten random movies and a search form. The user can rate the movies from 1 to 5 stars, according to his/her own personal tastes. See Figure 1.
- Then one by one the recommended movies are shown and the user has to compile the questionnaire in order to proceed with the recommendations.
- In the end, we ask the user to rate the experience overall; we also ask where s/he compiled the questionnaire and would s/he please add a general comment.
- Once the last answer is given, the questionnaire is promptly stored in the database and a csv file containing all questionnaires is automatically produced.
- When the administrator accesses the statistics page, the data is read and analyzed by the db, and graphs and tables are automatically produced. Almost all the graphs and tables shown in this report come directly from the website.



Your Choices:
2 more to go!

Mauvais sang
Rate: 4/5

The People vs. Larry Flynt
Rate: 4/5

The End of the Affair
Rate: 1/5

Year: 1953

Genre: Film-Noir, Thriller

Director: Samuel Fuller

Actors: Richard Widmark, Jean Peters, Thelma Ritter, Murvyn Vye

View more details

Suddenly

The Bug

Sayonara

Figure 1: Movies Choice

V. Metrics and procedure

There are two ways too analyze the quality of a recommender system, namely: analysis of objective metrics and analysis of user centered metrics. Objective metrics evaluate algorithmic quality, basing their data on repeated simulations and statistical evaluation, while user centered metrics evaluate the quality as perceived by the end user.

A. Objective Metrics

Objective metrics calculate the statistic probability of succeeding or failing the recommendation. There are various metrics for evaluating these qualities: *recall*, *fallout*, *precision*, *F-measure*, *RMSE*, *MAE*...

In our research we focused on recall and fallout, because some of the algorithms involved compute not actual ratings but only a probability prediction meanwhile, others need ratings of all the items for each user involved (an unrealistic condition in a real dataset such as ours). The used methodology is the same as [5]. Recall is defined as the conditional probability of suggesting a movie that is considered relevant to the user, the fallout is defined as the opposite (the condition probability of suggesting a movie irrelevant to the user). Everything is done within the URM data. For a series of lines in the URM the known ratings are split in two subsets, called test set and training set. The test set contains only 5-star ratings (or 1-star ratings for the fallout test set). The training set is the same for both the recall and fallout and contains a list of items rated by the user that will be used as the input for the recommendation. For each item in the test set, we join it to 1000 unrated items (we consider unrated items as irrelevant to the user) and use the training set to produce the recommendation list. We define hit the presence of the item from the 5-star test set in the first N (in our case, 5) positions in the list, and accordingly we define miss the presence of the item from the 1-star test set in the same ranks. We define recall as

$$recall = \frac{\#hits}{\#elements\,in\,T}$$

and fallout as

$$fallout = \frac{\#miss}{\#elements\,in\,T}$$

where T is the correspondent test set. See Table I to read riepilogative results.

1) Recall: Analyzing the results, we can say that the best algorithm in terms of recall is PureSVD, which has a recall of 0.17, directly followed by AsySVD (0.12). DirectContent has a value of 0.08 and, as expected, the lowest value is MovieAvg, with 0.04.

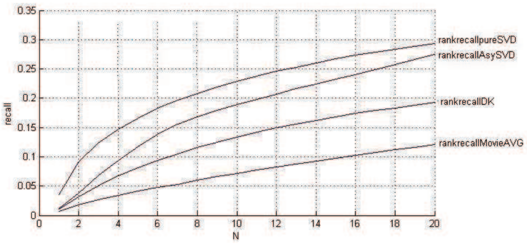


Figure 2: Recall

2) Fallout : The fallout shows a different rank between the algorithms than the one expected (the opposite of recall). The best algorithm (the one with the lowest fallout) is DirectContent (with the very low value of 0.005), while PureSVD and AsySVD show a very similar behavior with the value of 0.015. As before and as expected, the worst algorithm remains MovieAvg.

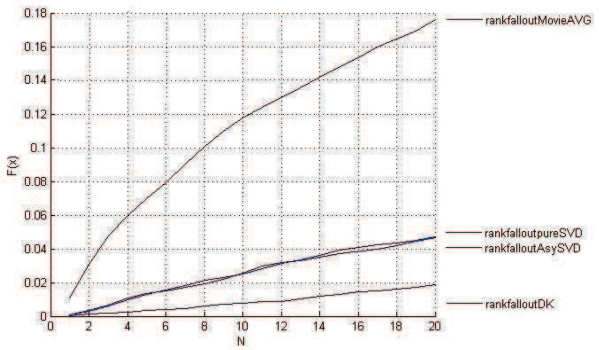


Figure 3: Fallout

Family	Algorithm	Recall	Fallout
Collaborative latent	AsySVD	0.12	0.015
Collaborative latent	PureSVD	0.17	0.015
Content	DirectContent KNN	0.08	0.005
Non-Personalized	MovieAvg	0.04	0.07

Table I: Objective Evaluation (top-5 recommendations)

B. User Centered Metrics

In the second part of our analysis, we focused on the user-end quality of the algorithms. In order to do this, we analyzed the answers to our questionnaire.

The parameters we used were the *perceived accuracy* (called also *relevance*), the *novelty* in its two orders (*First Order Novelty* and *Second Order Novelty*), and the *global satisfaction* [5].

In addiction to these metrics, we defined the *serendipity* (which measures the recommendation surprise) as the algebraic difference between the rate given after seeing the trailer and the one given before seeing it. In formula:

$$serendipity = r_{after\,trailer} - r_{before\,trailer}$$

C. Procedure and participants

Using the same interface and the same dataset, we asked the users to rate five movies s/he liked/disliked by choosing from one to five stars. We supplied the user with an initial list of ten random movies, as well as a search box enabling him/her to find a particular movie in the database. After each recommendation, the user was asked to answer a guided questionnaire. The questions were the same as those used in [5], based on the ResQUE model [8]. Each interview lasted between 5 and 15 minutes.

Our questionnaire followed the path described in Figure 2.

Also and above all, we tried to keep the algorithms distributed in a fashion as similar as possible.

VI. Data Analysis

In this section we analyse the results of the experiment and compare the two versions. You can see the detailed results in Table III and IV.

A. Relevance

1) Short Profile: In terms of average values, the best algorithm as to relevance is PureSVD, followed by DirectContent, AsySVD and MovieAvg.

2) Long Profile: The best algorithm as to relevance is PureSVD, followed by AsySVD, DirectContent and MovieAvg. All parameters show an increase in value. Box plot representation of both distributions can be found in Figure 5.

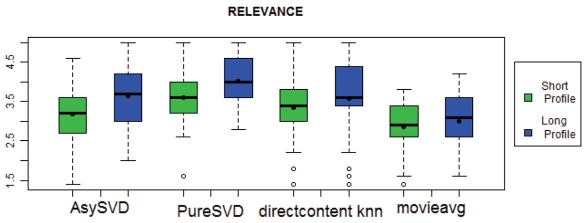


Figure 5: Relevance Box Plot

B. FON

1) Short Profile: The algorithm with the highest FON is DirectContent, followed by MovieAvg and AsySVD. PureSVD comes last due to its tendency to recommend popular movies (which the user is often familiar with).

2) Long Profile: The algorithm with the highest FON is MovieAvg, followed by DirectContent, AsySVD and PureSVD.

See Figure 6.

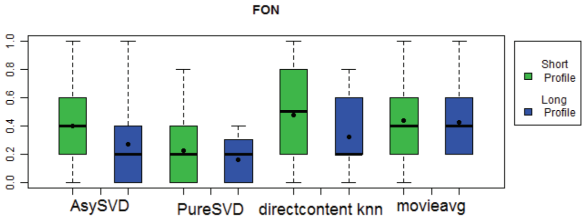


Figure 6: FON Box Plot

C. SON

1) Short Profile: SON, as expected, follows the same trend as FON.

2) Long Profile: The algorithm with the highest FON is DirectContent, followed by MovieAVG, AsySVD and PureSVD. Unlike with the 5 movies version, there's a (small) difference between FON and SON orders.

See Figure 7.

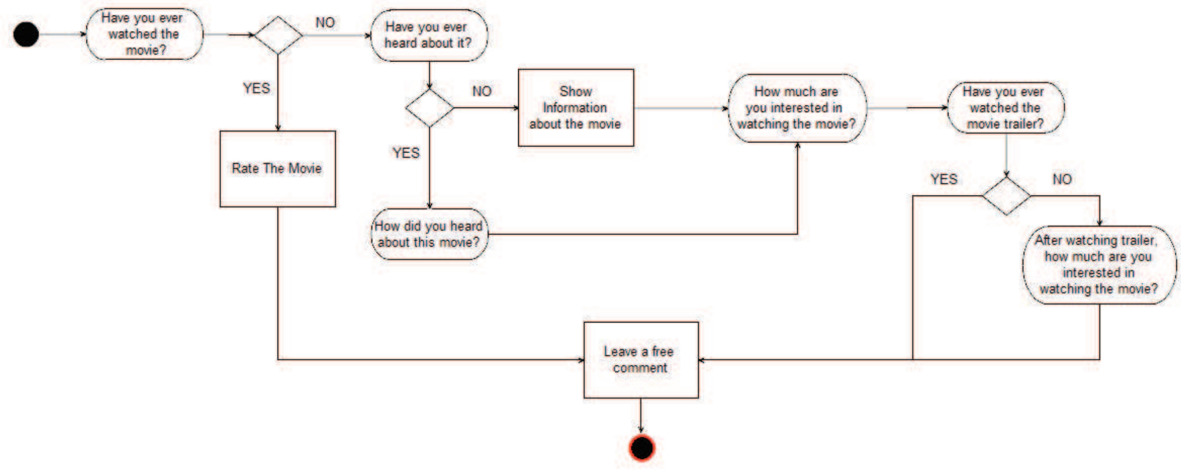

Have you ever watched the movie?
NO
Have you ever heard about it?
YES
NO
Show Information about the movie
How much are you interested in watching the movie?
Have you ever watched the movie trailer?
Rate The Movie
YES
How did you heard about this movie?
YES
NO
After watching trailer, how much are you interested in watching the movie?
Leave a free comment

Figure 4: Questionnaire

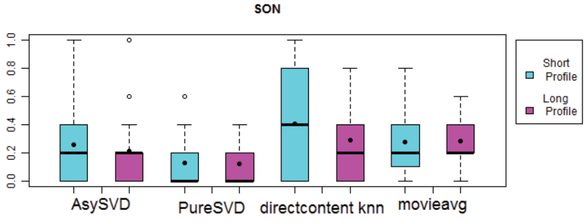


Figure 7: SON Box Plot

D. Serendipity

1) *Short Profile:* The algorithm with the highest serendipity is DirectContent, followed by PureSVD, MovieAvg and AsySVD.

2) *Long Profile:* The algorithm with the highest serendipity is PureSVD, followed by AsySVD, DirectContent and MovieAvg.

See Figure 8.

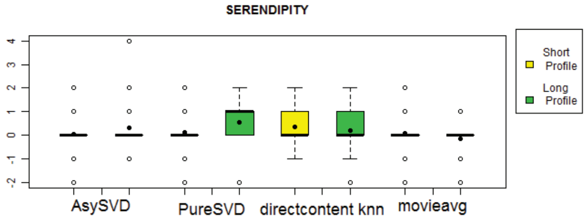


Figure 8: Serendipity Box Plot

We also performed a location test for this metric in order to test the difference between the mean value and 0. The result of the t-test are represented in this following table. The difference between 1 and the p-value represents the probability of being wrong saying that the average value is not 0.

Algorithm	*P-Value (Short Profile)*	*P-Value (Long Profile)*
AsySVD	0.6232	0.04802
PureSVD	0.4797	0.01212
DirectContent	0.002248	0.1817
MovieAVG	0.3776	0.2783

Table II: Location Test

E. Global Satisfaction

1) *Short Profile:* The algorithm with the highest global satisfaction is PureSVD, followed by AsySVD, DirectContent and MovieAvg.

2) *Long Profile:* The algorithm with the highest global satisfaction is PureSVD, followed by AsySVD and DirectContent (with the same value) and MovieAvg.

See Figure 9.

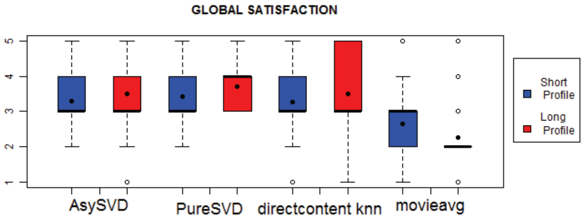
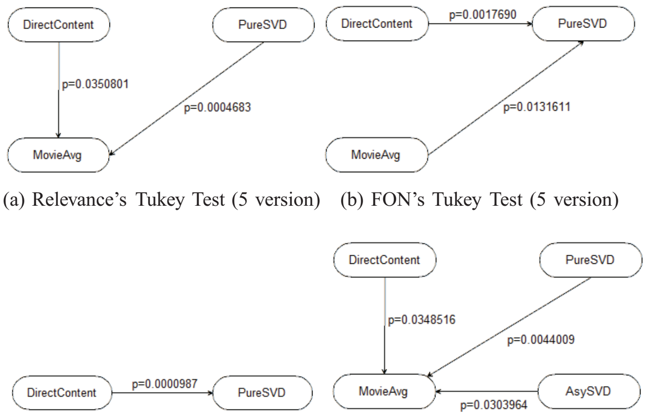


Figure 9: Global Satisfaction Box Plot

F. Siegel-Tukey's test (Between Algorithms)

In addition to these results, we were able to compare our results using Siegel-Tukey's test, a non-parametric test that may be applied to data measured at least on an ordinal scale. It tests for differences in scale between two groups. In our case, we used it to see if there were differences statistically important between the distributions of user-centered metrics. The test measures if the difference between two distributions has a mean value significantly far from zero. This means that the two distributions have a significative difference. To discriminate between the results, we see if the lower and upper values of the distribution (with a 95% tolerance) have the same sign. If not, the mean value is too close to zero and the results are disregarded. P-Value indicates the probability of being wrong in saying that the two distributions are different. Tukey's test lacks in transitivity, so ordering the algorithms in a hierarchical way without modifying the standard test procedure is impossible. The only comparisons we can analyze with certainty are shown in the graphs.

1) Short Profile: In terms of relevance DirectContent and PureSVD are significantly better than MovieAvg; in terms of FON DirectContent and MovieAvg are better than PureSVD; in terms of SON DirectContent is better than PureSVD; in terms of global satisfaction all algorithms are better than MovieAvg. No interesting comparisons were found in Serendipity. Detailed results are shown in Figure 8.

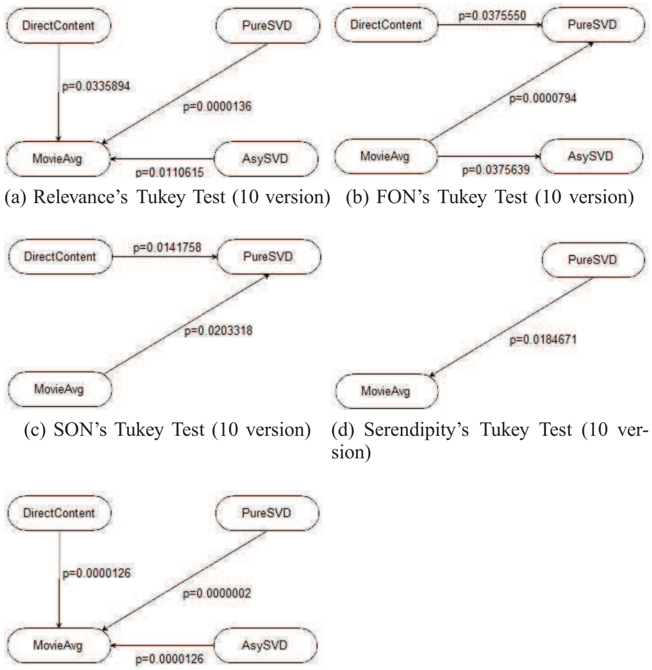


(a) Relevance's Tukey Test (5 version) (b) FON's Tukey Test (5 version)

(c) SON's Tukey Test (5 version) (d) Global Satisfaction's Tukey Test (5 version)

Figure 10: Tukey Test (5 version). Serendipity showed no significative result and is not represented.

2) Long Profile: In terms of relevance all algorithms are better than MovieAvg; in terms of FON DirectContent and MovieAvg are better than PureSVD, MovieAvg is better than AsySVD; in terms of FON DirectContent and MovieAvg are better than PureSVD; in terms of serendipity PureSVD is better than MovieAvg; in terms of global satisfaction all algorithms are better than MovieAvg. You can see the detailed results in the graphs in Figure 9.



(a) Relevance's Tukey Test (10 version) (b) FON's Tukey Test (10 version)

(c) SON's Tukey Test (10 version) (d) Serendipity's Tukey Test (10 version)

(e) Global Satisfaction's Tukey Test (10 version)

Figure 11: Tukey Test (10 version)

G. Location Siegel-Tukey's test (Between Short and Long Profile)

After completing the second experiment, we used Tukey's test to compare the results distribution between the two versions. In the twenty comparisons (four algorithms and five parameters) only three proved significantly different: AsySVD and PureSVD show an increase in relevance, DirectContent a decrease in FON. You can see detailed results in Table IV.

VII. Study Discussion

In this study we were able to test different kinds of algorithms in a unique, standard and extensible platform (TheBestMovie4You - http://www.moviers.it). This enabled us to add new features in a relatively simple way, namely an on-the-fly statistical analysis (almost all data in this document are taken directly from the website).

In particular, our data shows that in terms of relevance collaborative and content-based algorithms behave in a nearly identical fashion, while the non-personalized algorithms recommend items of little interest to the user. In our specific case, by reading the user comments we collected we realized that novelty is not always perceived as a good quality for an algorithm. This is because the user is frequently unfamiliar with and/or uninterested in the recommended items. For example, PureSVD has the highest relevance and the lowest FON (and SON) in the 5 movies version. In the case of DirectContent, the movies are often unknown to the user (it has the highest FON

Algorithm	Relevance	FON	SON	Serendipity	Global Satisfaction	Number of questionnaries
AsySVD	3.19	40%	26%	0.05 (40)	3.29	31
PureSVD	3.59	22%	13%	0.14 (22)	3.42	33
DirectContent KNN	3.35	48%	41%	0.35 (55)	3.26	34
MovieAvg	2.87	44%	28%	0.10 (41)	2.66	32

Table III: Average values for the user centered metrics in 5 movies version. Serendipity indicates in brackets the number of item on which it was calculated.

Algorithm	Relevance	FON	SON	Serendipity	Global Satisfaction	Number of questionnaries
AsySVD	3.65	27%	21%	0.32 (31)	3.50	30
PureSVD	4.02	16%	12%	0.55 (20)	3.71	31
DirectContent KNN	3.57	32%	29%	0.20 (35)	3.50	30
MovieAvg	3.01	43%	28%	-0.14 (44)	2.27	30

Table IV: Average values for the user centered metrics in 10 movies version. Serendipity indicates in brackets the number of item on which it was calculated.

Relevance					
Algorithm	*Difference*	*Lower*	*Upper*	*P-Value*	*Significative?*
AsySVD5-AsySVD10	**-0.4662366**	**-0.8847536**	**-0.0477195**	**0.029625**	**Yes**
PureSVD5-PureSVD10	**-0.4254154**	**-0.7435874**	**-0.1072435**	**0.0096017**	**Yes**
DirectContent5-DirectContent10	-0.2203922	-0.6674532	0.2266689	0.3282318	No
MovieAVG5-MovieAVG10	-0.1379167	-0.4663602	0.1905269	0.404274	No
FON					
Algorithm	*Difference*	*Lower*	*Upper*	*P-Value*	*Significative?*
AsySVD5-AsySVD10	0.1333333	-0.003180831	0.2698475	0.0554017	No
PureSVD5-PureSVD10	0.0629521	-0.04049683	0.166401	0.228428	No
DirectContent5-DirectContent10	**0.1564706**	**0.006399804**	**0.3065414**	**0.0412671**	**Yes**
MovieAVG5-MovieAVG10	0.01083333	-0.1111272	0.1327939	0.8595732	No
SON					
Algorithm	*Difference*	*Lower*	*Upper*	*P-Value*	*Significative?*
AsySVD5-AsySVD10	0.04473118	-0.07774966	0.167212	0.4678041	No
PureSVD5-PureSVD10	0.004692082	-0.08236442	0.09174858	0.914551	No
DirectContent5-DirectContent10	0.1192157	-0.03463807	0.2730694	0.1264883	No
MovieAVG5-MovieAVG10	-0.005	-0.102074	0.092074	0.9182831	No
SERENDIPITY					
Algorithm	*Difference*	*Lower*	*Upper*	*P-Value*	*Significative?*
AsySVD5-AsySVD10	-0.2725806	-0.6299649	0.08480363	0.1326879	No
PureSVD5-PureSVD10	-0.4136364	-0.968119	0.1408462	0.139491	No
DirectContent5-DirectContent10	0.1454545	-0.2094675	0.5003766	0.4175953	No
MovieAVG5-MovieAVG10	0.2339246	-0.09709259	0.5649418	0.1635852	No
GLOBAL SATISFACTION					
Algorithm	*Difference*	*Lower*	*Upper*	*P-Value*	*Significative?*
AsySVD5-AsySVD10	-0.2096774	-0.6516669	0.2323121	0.34636	No
PureSVD5-PureSVD10	-0.285435	-0.6791647	0.1082947	0.1523328	No
DirectContent5-DirectContent10	-0.2352941	-0.7901029	0.3195147	0.3998311	No
MovieAVG5-MovieAVG10	0.3895833	-0.07947595	0.8586426	0.1018555	No

Table V: Tukey's comparison between results of the two versions (Short - Long)

in the 5 movies version) yet consistent with his/her choices and interests.

Novelty is generally a good feature for an algorithm but often pays the price of recommending undesired items.

There's some kind of relation between relevance and global satisfaction (as shown in the relative scatter plot). Also, the order of algorithms is almost the same.

Our analysis shows that the number of movies chosen has a slight connection to the quality of the recommendations, as in only three cases out of 20 can we see a real improvement.

Overall, it's clear that algorithms reveal inner qualities different from one another. So, as shown here in the table (we reversed the order of fallout, being the only negative parameter), coming to a conclusion about the value of any single one is not easy. In Table V AsySVD appears as A, PureSVD as P, DirectContent as D, MovieAvg as M. We also indicate the relative version of our experiment.

VIII. Conclusions

In this study we demonstrated the lack of a relationship between user-centered metrics and objective metrics. We also pointed out that describing univocally the quality of an algorithm is not easy, for often it has particular features that makes it better than others.

Rank	Recall	Fallout	Relevance	FON	SON	Serendipity	Global Satisfaction
Maximum	P	D	5:P 10:P	5:D 10:M	5:D 10:D	5:D 10:P	5:P 10:P
Intermediate	A/D	P/A	5:D/A 10:D/A	5:A/M 10:A/D	5:A/M 10:A/M	5:P/M 10:A/D	5:A/D 10:A/D
Minimum	M	M	5:M 10:M	5:P 10:P	5:P 10:P	5:A 10:M	5:M 10:M

Table VI: Algorithm Qualities (User Centered metrics specify in which version of the experiment the order is obtained)

We must define a metric that can balance the weight of each of the user-centered metrics (Relevance, Novelty, Serendipity, Global Satisfaction), combining them into a single value in order to have a more reliable way to judge the behavior of a certain algorithm.

The website we developed is ready for expansion. It's really easy to activate other algorithms along with the four we analyzed, without re-implementing website functionalities or reconfiguration.

Like [5], we decided to use a short questionnaire rather than the long version proposed in the ResQue model in [8]. That way it's easier for the user to express an opinion about the recommendations instead of answering 60 questions (although the information taken from the complete questionnaire spans a wider set of parameters).

We focused our research on analyzing the main user centric metrics and now can say that it may be possible and advisable to keep moving forward in this direction. The automatic analysis of the data allows us to study a great number of questionnaires, so we can easily expand our work into other parameters (if automatically measurable).

The user did not perceive positively the non-personalized algorithm we chose, for in our dataset two of the five movies ("Anne Of Green Gables" and "Inuyasha") were evidently not famous films even though they did have a high average rate. (As explained earlier, MovieAvg cares not about the number of rates, only about their average value.) That's why our results for the non-personalized algorithm are different from [5] , because the user perceived these recommendations as non pertinent to his/her choices.

The comparison between the two phases of the experiment are somehow inconclusive: we get no clear improvement in preceived quality of the recommendation simply by increasing the selection of movies from five to ten.

IX. Future Developments

In the future we may find it advisable to expand the number of the interviews so as to collect a wider set of information. Another interesting idea may be to use the number of movies the user sees in one month as an index for measuring his/her appreciation level: if s/he sees just a few per month it may be advisable to recommend the blockbuster variety (using, for example, PureSVD and AsySVD), while if s/he is an authentic film buff it may advisable to use a more refined algorithm (such as DirectContent). We also feel a need to expand the movie dataset, as users often complained both about its limited size and about the lack of certain movies.

References

[1] J. L. Herlocker, "Evaluating collaborative filtering recommender systems." *ACM Trans. Inf. Syst. 22, 1*, pp. 5–53, 2004.

[2] S. M. McNee, J. Riedl, and J. A. Konstan, "Being accurate is not enough: how accuracy metrics have hurt recommender systems." *CHI '06 Extended Abstracts. ACM*, pp. 1097–1101, 2006.

[3] Z. Abbassi, S. Amer-Yahia, L. V. Lakshmanan, S. Vassilvitskii, and C. Yu, "Getting recommender systems to think outside the box." *Proceedings of the Third ACM Conference on Recommender Systems (New York, New York, USA, October 23 - 25, 2009). RecSys '09. ACM, New York, NY*, pp. 285–288, 2009.

[4] P. Cremonesi, F. Garzotto, S. Negro, A. Papadopolos, and R. Turrin, "Comparative evaluation of recommender system quality." *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems. CHI EA '11. ACM, New York, NY, USA*, pp. 1927–1932, 2011.

[5] ——, "Looking for good recommendations: A comparative evaluation of recommender systems." *Proceedings of the 13th IFIP TC13 International Conference on HumanComputer Interaction (INTERACT). Springer.*, 2011.

[6] O. Celma and P. Herrera, "A new approach to evaluating novel recommendations." *Proc. RecSys '08. ACM Press*, pp. 179–186, 2008.

[7] A. W. Shearer, "User response to two algorithms as a test of collaborative filtering." *CHI '01 Extended Abstracts . ACM*, pp. 451–452, 2001.

[8] L. Chen and P. Pu, "A user-centric evaluation framework of recommender systems." *ACM Conference on recommender systems (RecSys10)*, 2010.

[9] ——, "A cross-cultural user evaluation of product recommender interfaces." *Proc. of the 2008 ACM conf. on Recommender systems, RecSys '08*, pp. 75–82, 2008.

[10] R. Hu and P. Pu, "Acceptance issues of personality-based recommender systems." *Proc. of the third ACM conf. on Recommender systems*, pp. 221–224, 2009.

[11] P. Jones, N.and Pu, "User technology adoption issues in recommender systems." *Proc. of the 2007 Networking and Electronic Commerce Research Conf.*, pp. 379–394, 2007.

[12] P. Pu and L. Chen, "Trust building with explanation interfaces." *Proc. of the 11th int. conf. on Intelligent user interfaces, IUI '06*, pp. 93–100, 2006.

[13] P. Pu, L. Chen, and P. Kumar, "Evaluating product search and recommender systems for e-commerce environments." *Electric Commerce Research Journal*, 2008.

[14] P. Pu, M. Zhou, and S. Castagnos, "Critiquing recommenders for public taste products." *Proc. of the third ACM conf. on Recommender systems, RecSys '09*, pp. 249–252, 2008.

[15] J. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "The adaptive web." *Springer-Verlag, Berlin, Heidelberg, Chapter Collaborative filtering recommender systems*, pp. 291–324, 2007.

[16] G. Adomavicious and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-art and possible extensions." *Knowledge and data engeneering, IEEE Transaction*, pp. 734–749, 2005.

[17] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks." *RecSys '10: Proc. of the fourth ACM conf. on Recommender systems*, pp. 39–46, 2010.