# What is a Fair Value of Your Recommendation List?

Vladimir Bobrikov[1], Elena Nenova[1], and Dmitry I. Ignatov[2]

[1] Imhonet
vcomzzz@gmail.com, enenova@imhonet.ru
https://imhonet.ru
[2] National Research University Higher School of Economics
Moscow, Russia
dignatov@hse.ru

**Abstract.** We propose a new quality metric for recommender systems. The main feature of our approach is the fact, that we take into account the set of requirements, which are important for business application of a recommender. Thus, we construct a general criterion, named "audience satisfaction", which thoroughly describe the result of interaction between users and recommendation service. During the criterion construction we had to deal with a number of common recommenders' problems: a) Most of users rate only a random part of the objects they consume and a part of the objects that were recommended to them; b) Attention of users is distributed very unevenly over the list of recommendations and it requires a special behavioral model; c) The value of the user's rate measures the level of his/her satisfaction, hence these values should be naturally incorporated in the criterion intrinsically; d) Different elements may often dramatically differ from each other by popularity (long tail – short head problem) and this effect prevents accurate measuring of user's satisfaction. The final metric takes into account all these issues, leaving opportunity to adjust the metric performance based on proper behavioral models and parameters of short head problem treatment.

**Keywords:** recommender systems, quality metric, explicit feedback, movie recommendations, AUC, cold start, recommendations for novices

## 1 Introduction

Every recommender system aims to solve a certain business problem. Successful recommendations can be assessed in terms of specific business results, such as the number of visitors, sales, CTR, etc. However, it is too difficult to measure the quality of recommendation algorithm in this way since it depends on a vast variety of conditions, where the recommendation algorithm itself can bring a small contribution.

Therefore it turns out that developers need to come up with a formal numerical criteria for recommendation algorithms in isolation from the business

goals. As a result, a lot of papers on recommendation systems are produced every year. However, the numerical metrics they apply are useful, but usually are overly abstract compared to the problem they solve.

The approach we suggest is based on the idea that every metric should be constructed for a specific business problem. In this paper, we will focus on a concrete example, a movie recommendation service on `www.imhonet.ru`. Although we have tested the proposed approach on movies, this case can be generalized and applied to any similar objects (domain) of recommendation.

Let us shortly outline several relevant papers to our study. In [1] one of the most recent and consistent survey on evaluation of recommender systems can be found. Thus the authors discuss peculiarities of offline and online quality tests. They also review widely used quality metrics in the community (Precision, Recall, MAE, Customer ROC, Diversity, Utility, Serendipity, Trust, etc.) noting trade-off between these set of properties. Similar trade-off effects for top-$n$ recommendations were noticed and studied earlier [2]: "algorithms optimized for minimizing RMSE do not necessarily perform as expected in terms of top-N recommendation task". In [3], importance of user-centric evaluation for recommender systems through a so called user experiment is stressed; in fact, this type of experiments suggests an interactive evaluation procedure that also extends conventional A/B tests. In [4], the authors proposed a new concept of unexpectedness as recommending to users those items that different from what they would expect from the system; their method is based on the notions of utility theory of economics and outperforms baselines on real datasets in terms of such important measures as coverage, aggregate diversity and dispersion, while avoiding accuracy losses. However, the first approach which is close to our proposed metric is based on the usage of ROC curves for evaluation of customer behaviour and can be found in [5]; here, the authors modified conventional ROC curves by fixing the size of recommendation list for each user. Later, two more relevant papers that facilitated our findings appeared: 1) [6] continues studies with incorporation of quality measures (in the original paper, serendipity) into AUC-based quality evaluation framework and 2) [7] combines precision evaluation with a rather simple behavioral model of user's interaction with the provided recommendation list. In the forthcoming sections, we extend and explain how these concrete ideas can be used for derivation of our user-centric evaluation measure to fulfill business needs of the company.

The paper is organized as follows. In Section 2, we describe the main measured operation, i.e. the interaction between our service that provides recommendations and its consumers. It is important to list all the cases of possible types of interaction that our service can meet. Based on that cases, in Section 3, we substantiate the use of a common recommender's *precision* as a starting point of our inference. Then, in Section 4 we show how a common *precision* could be transformed into a stronger discounted metric even with the help of rather simple behavioral model. Section 5 is devoted to users' rates values; it describes how two different merits of metric, namely, the ability to evaluate a ranked list and the ability to be sensitive to rate values, could be joined in one

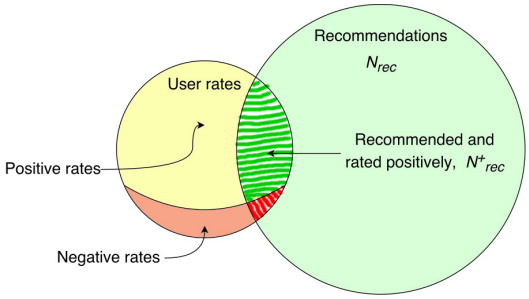


**Fig. 1.** Comparison of users' rates and recommended items.

term. In Section 6, we discuss how a *short head* problem could be treated. This specific recommender's problem makes it difficult to use those types of metrics that include sums of ratings. Section 7 summarizes the all previous considerations into the final expression for the metric and discusses several additional problems of its application. Section 8 demonstrates several illustrative cases of *Imhonet's* metric application from our daily practice. Section 9 concludes the paper and outlines future work.

## 2 Service product and users' feedback

The output of our service is a personalized list of recommended movies. The feedback comes directly from users in a form of movie rates. We shall start with the comparison of types of user's feedback, rates, and the lists of items that we recommend; they are shown in Figure 1. We need to consider the four situations.

**Hit** A movie has been recommended to a user, and it has received a positive rate from this user. This would normally mean that the recommendation was precise. We are interested in such cases, so let us call them "successes" and maximize their number.

**Mishit** A movie has been recommended but received a negative rate. We should avoid such situations. It is even possible that the avoidance of such cases is more important than maximizing the number of the cases of success. Unexpectedly, but we have learned from the experience that such cases can be ignored. The probability of the coincidence of negative signals with the elements from the list of recommendations, given by any proper recommender system, is too small to significantly affect the value of the metric. Therefore, it is not necessary to consider this case. This means that the metric is insensitive to negative rates.

**Recommended but not rated** If a movie has been recommended, but there has been no a positive signal, it seems that it does not mean anything. We do not know why it happened: a user has not seen the movie yet or has not rated it. As a result, it seems reasonable not to take into account these cases. The practice has shown that these cases constitute a majority. It happens due to two reasons. First, we always recommend a redundant number of movies, i.e.

more movies than a user could watch ($N_{rec}$ is relatively large). Second, most of the users tend to not give rates for every single movie they have seen, as if we could access only a fraction of users' rates.

**Rated but not recommended** It is the opposite case, the movie has not been recommended, but it has received a positive signal; hence, the recommender has not used an opportunity to increase the amount of successful recommendations. As long as these cases exist, it is still possible for the recommender to improve its efficiency. If all positively rated movies have been recommended, it means that the recommendation system's accuracy is the highest possible and there is no room for improvement.

## 3 Precision

If instead of just a number of successes we use the value of *precision (p)*, i.e. we divide the number of successes $N^+_{rec}$ by $N_{rec}$, there will be no a significant change: instead of the number of successes we will maximize the same value, only divided by a constant:

$$p = \frac{N^+_{rec}}{N_{rec}}. \tag{1}$$

However, as we will see later, this division provides an opportunity to make the metric sensitive to a very important aspect of our problem. (It allows us to make it discounted, in other words, – to take into account the order of the elements in the recommendation list.) Moreover, the value of $p$ has a clear meaning, which can be described in a probabilistic language. Assume that a user consumes our product, namely, go through all the elements of the recommendation list. Then $p$ shows the probability for him to find in this list a suitable element, i.e. the one that will satisfy him in the future. Denote *precision* for the user $u$ as $p_u$:

$$p_u = \frac{N^+_{rec}(u)}{N_{rec}}. \tag{2}$$

Now we can generalize this formula for our entire audience (or a measured sample) of *Users*:

$$P_{N_{rec}} = \underset{u \in Users}{mean}(p_u) = \frac{1}{|Users|} \cdot \sum_{u \in Users} \frac{N^+_{rec}(u)}{N_{rec}} = \frac{N^+_{rec}}{N_{rec} \cdot |Users|}. \tag{3}$$

Every user looks through his own list and chooses what he/she needs, so $P_{N_{rec}}$ shows the average probability of success for all occasions. The value on the right side is the total number of successes in the whole sample.

## 4 Discounting

So far we have evaluated a list of elements as a whole, but we know that its head is more important than the tail – at least, if the list is not too short. The

metric, that takes this into account and, therefore, depends on the order of the elements in the list, is called a *discounted metric*.

The list's head is more important than the tail due to the uneven distribution of user's attention: people more frequently look at the first element, they less frequently look both at the first and the second elements of the list, etc. This means that a proper discounting requires a behavioral model and the data that can support and train this model.

Let us imagine an arbitrary user who looks through the elements of the list one by one, starting with the first element, then the second, the third... and then, at some point, stops. There is no need to identify a specific user, because sometimes the same person wants to see the list of 2 elements, and sometimes the list of 20. It might be useful to know the average probability of transition from one element to another, but we do not need such precise data. If there is a probability $w_N$ that an arbitrary user goes through a list of $N$ elements for any plausible $N$, then we can average the value of $P_N$ according to the law of a total probability, where $P_N$ is the average probability of success for the part of our audience, that went through the list of $N$ elements. It can be described by the following definition:

$$AUC = \sum_{N_{rec}=1,\ldots,\infty} w_{N_{rec}} \cdot P_{N_{rec}} \tag{4}$$

In this definition $N$ was replaced with $N_{rec}$. It turns out that in contrast to precision, $AUC$ value estimates the average probability of success of personal recommendation lists in a real life environment, when the users' attention is unevenly distributed. Note that in order to derive the value of the $AUC$ we used the dependence of precision $P_{N_{rec}}$ on the size of the recommendation list $N_{rec}$, which was considered as fixed earlier.

Let us note that the term $AUC$ is also used to represent the *precision by recall integral*, which is sometimes used as a quality metric for classifiers [8]. The sum we calculated in Formula 4 is an analogue of this metric: different $N_{rec}$ values simulate different Recall values.

### 4.1 An easy way to estimate $w_N$ values

There is a simple evaluation model for $w_N$, which allows not to handle all of the transition probabilities, but provides a qualitative description of user's behavior. The only model parameter is the probability $Q$ that an arbitrary user moves to the second page in the list, which is available through pagination web logs. Assume that each page contains $m$ elements and users proceed to the next viewing element with the same probability $p$ (or leave with the probability $(1-p)$). Then $p$ can be easily obtained from the $Q = p^m$ ratio, assuming that the first element is viewed with a probability of 1. Then, the probability $w_N$ that a user sees $N$ elements and then stops can be easily calculated with the following equation:

$$w_n = p^{(N-1)} \cdot (1-p). \tag{5}$$

A similar approach, where transition probability $p$ is set to be a constant was used in [7].

## 5 Rate value and satisfaction

So far we have only been using positive signals while ignoring the fact that we have their values. Clearly, it will be unreasonable to neglect this data. If the rate scale is tuned well, the rate value will be, on average, proportional to the user's satisfaction level. Taking into consideration the above, we can try to replace the counter of successes $N^+_{rec}$ in Equation 4:

$$AUC = \sum_{N_{rec}=1,..\infty} w_{N_{rec}} \cdot P_{N_{rec}} = \sum_{N_{rec}=1,..\infty} w_{N_{rec}} \cdot \frac{N^+_{rec}}{N_{rec}|Users|} \tag{6}$$

with more informative sum of the positive rates:

$$AUC^r = \sum_{N_{rec}=1,..\infty} w_{N_{rec}} \cdot \frac{1}{N_{rec}|Users|} \cdot \sum_{r \in S_{N_{rec}}} (r-5), \tag{7}$$

where $S_{N_{rec}}$ is the set of successful rates, i.e. the positive rates which were counted in $N^+_{rec}$. On our 10-stars scale we consider six stars and more as a positive rate, so for the convenience we subtract 5 from all rates (implying that we sum up only the "satisfaction stars").

The role of the positive rates in the metric can also be represented in a different way:

$$AUC^r = \sum_{N_{rec}=1,..\infty} w_{N_{rec}} \cdot P_{N_{rec}} \cdot r^+_{mean}(S_{N_{rec}}), \tag{8}$$

where

$$r^+_{mean}(S_{N_{rec}}) = \frac{1}{N^+_{rec}} \cdot \sum_{r \in S_{N_{rec}}} (r-5). \tag{9}$$

We can think about the last term, that it is an average rate (positive and successful) among the audience that went through the list of $N_{rec}$ elements. The product of the success probability and an average positive rate in case of success could be described as the total *satisfaction level*, that can be provided by the recommendation algorithm. In this way the metric, although losing a purely probabilistic interpretation, is now better suited for our purposes.

## 6 Long tail and short head

Amusingly, when it comes to movies and other media products, the popularity distribution of the elements is extremely uneven. The number of movies that are well-known and has been rated by a large amount of people is very small, it is a *short head*. The vast majority of movies stay unknown to the audience, it

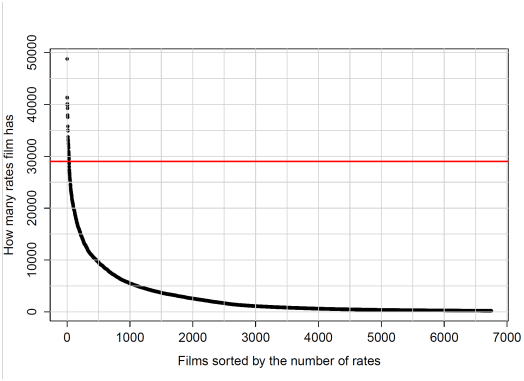


**Fig. 2.** Rates distribution on Imhonet.ru.

is a *long tail*. For example, the Imhonet rates distribution looks like the one in Figure 2.

The short-head rates distribution curve is so steep, that any rates summation will lead to the short-head movies domination, giving an 80% contribution to the metric. This causes its numerical instability: appearance or disappearance of a few short-head objects in the top of recommendation list can dramatically change the value of the metric.

Let us not forget that the goal of recommendation system is an effective personalization. It is primarily associated with the ability to select items from the long tail, because the elements of the short head are familiar to everyone, so there is no need to include them in the recommendation.

Therefore it seems reasonable to reduce a *short head* weight in the metric. In order to do it correctly, why do not make a start from the metric problem, which is the fact, that the numerical instability reduces sensitivity. We model the test cases for the metric to distinguish and try to achieve its maximum sensitivity.

As a starting point we take the situation when we know nothing about the users in our sample. In that case all personal lists of recommendations reflect an average movie rating and hence look exactly the same. In order to construct this rating we can use a simple probability, based on an average rating, that the movie will be liked. For example, it may be the probability, that an average movie score is higher than five points:

$$P(r > 5) = \frac{|The\ movie\ rates\ greater\ than\ 5|}{|All\ the\ movie\ rates|} \tag{10}$$

Fortunately, in addition to the rates we have a lot of different information about the users from the questionnaire: gender, age, preferences, interests, etc. For example, if we know the gender, we can build two different recommendation lists instead of a generalized one. It can be done using Bayes' formula:

$$P(r > 5|man) \propto P(man|r > 5) \cdot P(r > 5) \tag{11}$$

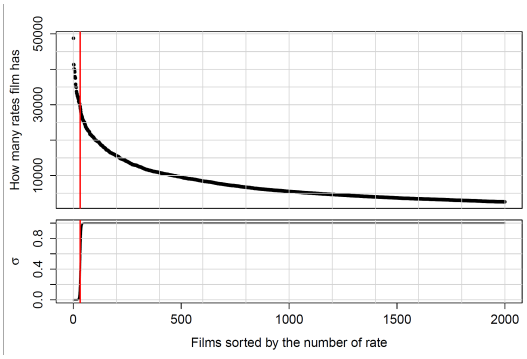


**Fig. 3.** The *short head* and sigmoid penalty function.

$$P(r > 5|woman) \propto P(woman|r > 5) \cdot P(r > 5) \tag{12}$$

Here, the probability for our starting point $P(r > 5)$ works a priori. Since two different recommendation lists are better than one, we can expect growth in the value of the metric.

It is more convenient to evaluate the relative increase:

$$\frac{AUC(man/woman) - AUC_0}{AUC_0} > 0. \tag{13}$$

The increase of the metric will take place every time we use any additional user information, essential for the users' preferences of movies. The more metric increase, the more it is sensitive to the information. Since we are not specifically interested in gender, in order to avoid over-fitting on this particular case, we will average AUC increase based on the variety of the criteria we use to segment the audience. In our experiment we used the users' answers to a 40 questions questionnaire.

Let us move on to an optimization problem. It can be described as searching for the metric with the best sensitivity. As you remember, the basic solution of the short-head/long-tail problem is to reduce the weight of the short-head elements. We denote the function responsible for the short-head elements penalties as $\sigma$. The $\sigma$-function must provide a maximum sensitivity:

$$\underset{\sigma}{argmax} \left( \underset{g \in G}{mean} \left( \frac{AUC_g(\sigma) - AUC_0(\sigma)}{AUC_0(\sigma)} \right) \right), \tag{14}$$

where $G$ is the set of audience segmentations with relevant recommendations for each segment. In a simple experiment, which proved to be effective, we have used a step function $\sigma$ (approximation of sigmoid) to null the *short head* elements weight as it is shown on the Figure 3. This means that the optimization problem 14 needs to be solved for a single parameter $\sigma$, which determines the position of the step in the list of movies, sorted by the number of their rates.

## 7 Final formula

Here is the final formula of the metric, that takes into account all the above reasoning:

$$AUC^r = \frac{1}{|Users|} \cdot \sum_{N_{rec}=1..z} \frac{w_{N_{rec}}}{N_{rec}} \cdot \sum_{r_{ui} \in S_{N_{rec}}} \sigma(i) \cdot (r_{ui} - 5), \qquad (15)$$

- $N_{rec}$ is the length of recommendation list;
- $z$ is the *precision* values summarizing limit. For long lists precision values are very small, as well as multiplier $w_{N_{rec}}$, so significantly large $z$ value does not affect the metric;
- $|Users|$ is the number of users in the sample;
- $w_{N_{rec}}$ is the probability that a random user will look through a list of $N_{rec}$ elements exactly;
- $S_{N_rec}$ is the number of positive rates in the recommendation list of $N_{rec}$ elements;
- $\sigma(i)$ is the penalty function value for an element $i$;
- $r_{ui}$ is the rate of the movie $i$ received from the user $u$.

## 8 Experiments

In this part we will discuss some practical examples of the metric application[3]. We have used a set of special machine learning models of *imhonet.ru* recommendation system. These models are not described here in greater details, since we only want to illustrate using the metric.

*Cold start* is one of the most crucial problems for recommendation system. There are two kinds of the cold start problem: *a new user* and *a new element*.

### 8.1 New users

Let us compare the quality of recommendations based on an arbitrary user rates along with the quality of recommendations based on the additional information about the user. The latter recommendations can be designed in order to solve the cold start problem by the following methods:

1. Finding out user's age and gender;
2. Giving a user few simple questions to answer;
3. Using user‘s rates given to non-movies elements.

The results of metric calculation for all these methods are presented in Figure 4. The black horizontal line at the bottom of the plot represents the quality of recommendation list in case we have no information about users and suggest all of them the same list of recommendations calculated by 10.

[3] All the datasets used in the experiments are available from the first author of this paper by e-mail request

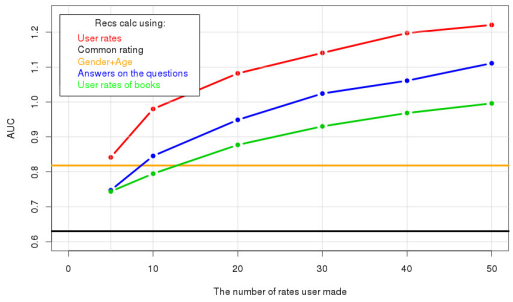


**Fig. 4.** Decision of new-users problem.

The topmost red curve shows the dependence of recommendations quality from the number of objects rated by user. The more objects user rates, the more precise his/her list of recommendations is.

The orange horizontal line near 0.8 *AUC* level shows the quality of recommendations in case we know only gender and age of the users. Information about user's gender and age makes the metric more precise as much as about 5 rates.

Now consider the blue curve under the red one. For a common user it is not always easy to put their preferences directly into rates, so we offer newcomers few simple questions, such as "Do you love anime?" or "Do you like action?", that are chosen from a few hundred questions list. Although questions are not as informative as rates (for example, 10 rates are equivalent to 25 answers), they are still useful, since for the majority of users it is easier to answer a question rather than to give numerical rate.

Let us explain the lowermost green curve. Sometimes we have to deal with the users who has already got a profile, based on the rates of the elements from non-movie domains. If there is a connection between preferences in different domains, we can try to use it. In our case, the users have already got profiles in fiction books section, and we are trying to use this information in order to recommend them some movies.

### 8.2 New items

It is important to be able to recommend new elements before they have received enough rates from users. Clearly, this can only be done on the basis of information about the movie itself. In our recommendation system movie properties that have a key influence on the recommendations are as follows: *genre*, *director*, *actors*, *screenwriter* These metadata make it possible to recommend a movie as if "experienced" users (not newcomers) have already given it about 27 ratings, which you can see in Figure 5.

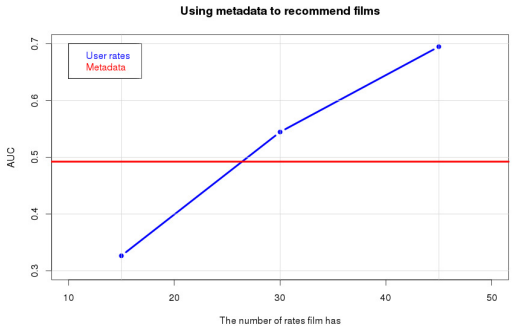


**Fig. 5.** Solution of the new-items problem.

A similar problem was described in [9]. Their *SVD*-based recommendation models for new elements were evaluated by RMSE. 10 user rates appeared to be more valuable than the metadata description.

## 9 Conclusion and future work

In this paper we have described the consistent inference of the metric for a recommendation system which is based on explicit users' rates and designed to provide a ranked list of recommended elements. We hope that the metric possesses a set of specific properties, such as: it is sensitive to the order of the items on the list or, more precisely, discounted in accordance with a simple behavioral model, when the user goes through the recommendations one by one, from the top to the bottom; it takes into account the value of positive ratings, so it can measure not only the concentration of successes, but also the amount of satisfaction; correctly handles short head/long tail problem — penalizes short head elements to optimize the sensitivity;

The main purpose of the metric inference is to develop an effective tool, that could be used for the recommendation algorithm optimization accompanied by the improvement of the business metrics. It means that in order to estimate the metric efficiency we will have to compare the target business metrics with the dynamic of the recommendation system metric. Although, as we have noticed in the introduction, this procedure is quite complicated and will be discussed further later.

**Acknowledgments** We are grateful to Prof. Peter Brusilovsky (University of Pittsburgh), Prof. Alexander Tuzhilin (NYU/Stern School), and Prof. Alexander Dolgin (HSE, Moscow, Russia) for discussions and helpful suggestions. We also would like to thank participants of clasess on recommender systems at NewProLab and HSE, Moscow, as well Sheikh Muhammad Sarwar (Dhaka University,

Banladesh). The third co-author was partially supported by Russian Foundation for Basic Research, grants no. 16-29-12982 and 16-01-00583.

## References

1. Gunawardana, A., Shani, G.: Evaluating recommender systems. In: Recommender Systems Handbook. (2015) 265–308
2. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010. (2010) 39–46
3. Knijnenburg, B.P., Willemsen, M.C.: Evaluating recommender systems with user experiments. In: Recommender Systems Handbook. (2015) 309–352
4. Adamopoulos, P., Tuzhilin, A.: On unexpectedness in recommender systems: Or how to better expect the unexpected. ACM Trans. Intell. Syst. Technol. **5**(4) (December 2014) 54:1–54:32
5. Schein, A.I., Popescul, A., Ungar, L.H., Pennock, D.M.: CROC: A new evaluation criterion for recommender systems. Electronic Commerce Research **5**(1) (2005) 51–74
6. Lu, Q., Chen, T., Zhang, W., Yang, D., Yu, Y.: Serendipitous personalized ranking for top-n recommendation. In: 2012 IEEE/WIC/ACM International Conferences on Web Intelligence, WI 2012, Macau, China, December 4-7, 2012. (2012) 258–265
7. Moffat, A., Zobel, J.: Rank-biased precision for measurement of retrieval effectiveness. Trans. Inform. Syst. 27 (December 2008)
8. Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: ICML '06 Proceedings of the 23rd international conference on Machine learning. (2006) 233–240 ISBN:1-59593-383-2.
9. Pilászy, I., Tikk, D.: Recommending new movies: even a few ratings are more valuable than metadata. In: Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009, New York, NY, USA, October 23-25, 2009. (2009) 93–100