. JOURNAL OF COMPOUTER SCIENCE AND TECHNOLOGY : 1– Mon. Year

Performance Characterization of Game Recommendation Algorithms on Online Social Networks Sites

Philip Leroux[1], *Student Member, IEEE*, Bart Dhoedt[1], *Member, IEEE*, Piet Demeester[1], *Fellow, IEEE* and Filip De Turck[1], *Senior Member, IEEE*

[1]*Ghent University - IBBT - Department of Information Technology, G. Crommenlaan 8/201, B-9050 Gent, Belgium*

E-mail: {Philip.Leroux, Bart.Dhoedt, Piet.Demeester, Filip.DeTurck}@intec.ugent.be

Received month day, year

Abstract Since years, online social networks have evolved from profile and communication websites to online portals where people interact with each other, share and consume multimedia-enriched data and play different types of games. Due to the immense popularity of these online games and their huge revenue potential, the number of these games increases every day, resulting in a current offering of thousands of online social games. In this paper, the applicability of neighborhood-based collaborative filtering (CF) algorithms for the recommendation of online social games is evaluated. This evaluation is based on a large data set of an online social gaming platform containing game ratings (explicit data) and online gaming behavior (implicit data) of millions of active users. Several similarity metrics were implemented and evaluated on both the explicit data, implicit data and a combination thereof. It is shown that the neighborhood-based CF algorithms greatly outperform the content-based algorithm, currently often used on online social gaming websites. The results also show that a combined approach, i.e. taking into account both implicit and explicit data at the same time, yields overall good results on all evaluation metrics for all scenarios, while only slightly performing worse compared to the strengths of the explicit or implicit only approaches. The best performing algorithms have been implemented in a live setup of the online game platform.

Keywords Mining methods and algorithms, Data mining, Personalization

1 Introduction

According to research released by The Nielsen Company in 2010 [1], Americans spend 40 percent of their time online on just three activities: social networking, playing online games and emailing. Nearly a quarter of their time online is spent on social networking sites and blogs, while online games overtook personal email to become the most heavily used activity after social networks, accounting for 10% of all U.S. Internet time (against 8.3% for emailing). According to this research, Facebook is the most popular destination for online games as 83% of the American respondents play their online games on this popular social network site.

While still in its infancy compared to the traditional game industry, the social game industry represents a huge opportunity to reach hundreds of millions of users who often have never played video games before. The major

companies in this rapidly rising industry are social game developer companies such as Zynga, Playfish or PopCap whose titles like FarmVille, Zynga Poker, Restaurant City or Bejeweled are played by millions of users each day. For instance, as of May 2011, Zynga, the biggest player in the market has close to 250 million monthly active Facebook users, with CityVille having over 20 million daily active users. The games themselves are most of the time built around the microtransactions model: a free-to-play business model that relies on users paying small amounts of money for virtual goods and in-game items to enhance their game play. A March 2010 report by European investment bank GP Bullhound [2] stated that the global social gaming sector made US$1 billion in revenue in 2009 and this number is expected to rise to US$3 billion by 2012.

With so much revenue potential in social gaming, it is not surprising that more companies are trying to reach those millions of potential online customers, by releasing their own online social game titles. This results in such a huge offering of online games that users typically find it difficult to select the most appropriate game and thus often will rely their choice based on their social connections. A survey [3], unveiled by PopCap Games, found that word-of-mouth is the most common way that social gamers hear about new social games. 57% of online social gamers rely on a recommendation or in-game alert from a friend while 38% allegedly learns about new games from ads on social networking sites. Finally, 27% cited standard Web searches as a source of information on new social games. As social gamers tend to play their favorite games with great frequency, it is of main importance for both the hosting social website as the game companies that the most appealing game for a specific user actually is known and played by that user as this may result in an even more increased number of game plays and revenues.

To date, a variation of recommendation techniques have been developed. Broadly speaking, they are all based on two different strategies or a combination thereof. In the *Content-Based approach*, a profile is created for each user or product item, describing the characteristics of the user or product. For example, the profile of an online game could include attributes regarding its game category (e.g. shooter, strategy, sports), the developer company, release date, etc. while user profiles very often consist of mainly demographic information. Recommendations are then made by associating the profiles of users with those of the product items.

The second strategy, known as *Collaborative filtering (CF)* relies on the past user behavior and how this behavior relates to that of other users. Such a behavior related profile can be created by taking into account explicit and/or implicit ratings. Explicit ratings are ratings where users assign a value to an item based on an agreed rating scale. For example, Netflix collects star ratings for movies while Youtube users indicate their taste by hitting the thumbs-up or thumbs-down button. On the other hand, implicit ratings are inferred from the users' actions (e.g. gaming, shopping, requesting information) and don't require the user's intervention. While implicit information may often contain noisy information, its use is often compulsory in many practical situations, often due to the lack of sufficient explicit data.

In this paper, collaborative filtering recommendation algorithms are evaluated on large scale social gaming data. During the evaluation of all experiments, both implicit and explicit data together with the game profile data were extracted from an obfuscated data set gathered by Gatcha![4], a European game developing and game hosting company with millions of active gamers, and sister company to the popular online social network site Netlog [5]. This data set is described in Section 3. The

paper is organized as follows: in Section 2 related research is described. Section 3 describes the data set that was analyzed in detail during the experiments. Section 4 provides the details of our methodology, describing the data processing, the algorithm design, the experimental setup and the evaluation methodology. Section 5 presents the obtained evaluation results and finally section 6 states our conclusions and future work.

2 Related Work

To date, a variation of recommendation techniques have been developed and of these, collaborative filtering (CF) techniques have been one of the most successful. The main advantage of these techniques lies in the fact that they can deal with any kind of content and recommend any type of items, even ones that are dissimilar to one another. Although CF systems have many forms, the basic idea of all CF algorithms is to recommend items to a target user by predicting their utility for that user through previous ratings by other users.

In literature, many CF systems only take explicit ratings into account. For e.g., Grouplens [6] recommends news articles to users based on their previous ratings on articles in Usenet. In the last years, CF algorithms based on explicit data, gained even more interest with the start of the Netflix prize in 2006 [7]. Netflix, an online DVD-rental service, offered a grand prize of US$1 million for the best CF algorithm to predict user ratings for movies, based on their previous ratings. Two of the biggest challenges of the competition were both the sparsity and size of the dataset. This competition ended in 2009 and formed the basis for many research papers.

In many practical recommender systems, it is often difficult to obtain explicit feedback. However, interesting information may also be inferred from analyzing user behavior or tracking user actions. The difficulty with implicit feedback is that it is inherently noisy. While we passively track the users behavior, we can only guess their preferences and true motives. In recent years, and especially with the integration of many CF systems into web applications, several academic and industrial projects have been focusing on the deduction and translation of user behavior into personal scores, called implicit ratings. E.g. in [8] a framework is presented for mining user preferences from reviews and then mapping such preferences onto numerical rating scales. In [9], the user's online media sharing activities are used as a source of implicit feedback for a recommendation system and in [10], implicit links are drawn between Web pages to improve the Web page classification performance. An alternative approach to the rating prediction approach is the ranking-oriented approach. In this approach, recommender systems focus on the prediction of the correct ranking of missing items without the intermediate step of predicting the ratings of these missing items. Research [11] [12] has shown that focusing on the ranking of items can lead to better recommendations.

Very often, recommender systems use a combination of techniques or data to generate a final recommendation score. In [13], matrix factorization models were developed that can be trained from explicit and implicit feedback simultaneously. Social networks and recommender systems are combined in FilmTrust [14], a website that uses trust in web-based social networks to create predictive movie recommendations. Similarly, in [15], users are first clustered based on the trust between them using correlation clustering and then a collaborative filtering algorithm is optimized by using these clusters. A similar approach is applied in [16], where clusters for users, Web pages and social annotations, describing the Web resources, are first generated. Web page recommendations are then generated based on the trans-

actional information of the users, Web pages, social annotations and the acquired cluster information.

Although there are (general) recommender systems that also encompass the recommendation of game products (e.g. on Amazon) there have been no previous publications that specifically focus on the recommendation of online games based on the gaming behavior and personal game ratings of users, as described in this paper. As detailed in the Data Set Description of Section 3, the users' game rating behavior was found to contain rather extreme values (due to users mainly awarding games with either very high or very low star ratings) in comparison to the rating behavior for other use case scenarios. While research [17] has shown that correlated features in an unbalanced data set may be found by applying association rules, the work presented in this paper focuses on the addition of implicit data to the rather unbalanced explicit data set to improve the recommendations. This extreme rating behavior, in combination with the analysis of gaming behavior over a longer period of time, makes this evaluation and optimal parameter configuration for several established recommender algorithms very relevant. Finally, within the scope of game-specific research, most research is performed within the field of dynamic gameplay adaption [18, 19].

3 Data Set Description

3.1 Explicit Data

The experiments in this paper are all based on the data set of the online game distribution platform Gatcha! [4]. This platform enables game developers to deploy their game onto several social network sites and web-portals. The explicit dataset contained nearly 2 million explicit ratings for more than 200 different games, made by almost 6 million unique users spread over 260 different countries. These explicit ratings were star ratings on a scale from 1 to 10. The exact spread of these ratings is shown in Figure 1.

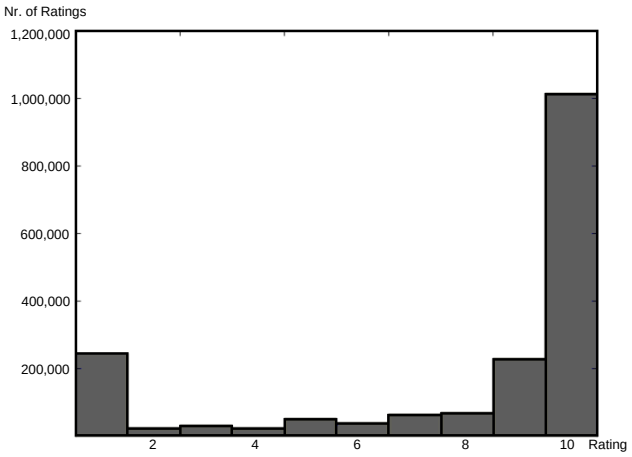


Fig. 1. Distribution of the explicit game ratings (on a scale from 1 to 10), based on a data sample of 1.8 million ratings.

As Figure 1 shows, the rating behavior of gamers is rather extreme, awarding games with either very high or very low star ratings, depending on whether they like or dislike a game. The distribution also indicates that users only tend to rate a game when they really like the game as the 10 star ratings take almost 60% of the total number of ratings into account. A second trend that was shown whilst analyzing the explicit game ratings, was that users that did rate a game, usually do this for more than one game. This is shown in Table 1.

Table 1. Distribution of the users based on the number of games they have rated in the data set.

Nr. of Ratings	Users
1 to 5	29,391
5 to 10	415,899
10 to 15	137,056
15 to 20	71,300
20 to ...	49,660

As shown in Table 1, most users that have rated at least one game, usually did this for 5 to 10 games. Note that the data sample of explicit ratings was taken over a period of almost 2 years, taking into account all ratings of those users that were evaluated. However, with respect to the rating behavior of users, the most important conclusion is that only 12% of all users in the data sample have explicitly rated at least one item.

In the field of recommendation techniques, and especially with respect to neighborhood-based CF algorithms, the sparsity of explicit data is found to be a common problem [20], especially as the incentive for users to rate a (consumed) item is often very low. The implicit rating approach attempts to increase the number of ratings entered by observing user behavior.

3.2 Implicit Data

The implicit ratings were inferred from monitored game plays of 1.5 million users over a period of almost six months (173 days exactly) resulting in a total number of more than 126 million single player game plays and almost 21 million multi-player game plays. From each game play also the start and end time were provided. The latter was only recorded for users normally ending the game, not taking into account sessions ending by closing the browser before actually closing the game. As shown in Table 2, most players played during the sample period typically 1 to 5 games, with a major share of people only playing 1 specific game. The maximum number of games played by one user was 138. With respect to the number of single player game plays, similar conclusions can be drawn, with a major share of people residing in the lowest category of 1 to 5 game plays. However, the share of game plays in this category is more equally distributed and the total share is also a little less predominant compared to the played games scenario.

Table 2. Distribution of the users based on the number of single player games they have played (column 1 and 2) and the number of single player game plays they have performed (column 3 and 4) over a period of 6 months.

Played Games	**Users**	**Game Plays**	**Users**
1 to 5	1,133,210	1 to 5	796,249
5 to 10	167,604	5 to 10	178,441
10 to 15	64,313	10 to 20	125,470
15 to 20	32,716	20 to 50	118,850
20 to 30	29,988	50 to 100	67,662
30 to 40	11,486	100 to 500	111,089
40 to 50	4,922	500 to 1000	26,637
50+	3,649	1000+	23,490

The advantage of this data set is that it allows to evaluate users' total gaming behavior over a relatively long period and that the length or date of each game play may be taken into account. However, it does not provide any details about the games that were played by the users before or after the period of the data sample. Section 4.1 elaborates on this problem.

4 Methodology

4.1 Processing Game Data

During the experiments, the same algorithms were executed on both the implicit and explicit data (or a combination of the two). In order to provide a consistent rating domain in comparison with the implicit ratings, detailed in the following section, the explicit star ratings, ranging from 1 to 10 stars, were rescaled within the [0,1] interval.

With respect to the implicit data, research [21] shows that the unique characteristics of implicit feedback, prevent the direct use of algorithms that were designed with explicit feedback in mind. These main characteristics are:

no negative feedback, noisy data, indication of preference instead of confidence and the necessity of proper measures for the evaluation. These remarks are valid for most types of data as recommender engines for e.g. digital TV systems or product websites may presume that when a user consumed or bought a specific item, the user was interested in the item but they are often not able to infer whether that user actually liked the item when no explicit rating was given afterwards. Or they are not aware that a specific item was bought as a gift. However, for the game specific data it can be stated with high confidence that the number of game plays of a game by a specific user is directly coupled with that user's interest for that specific game. This also implies that negative feedback can be taken into account, e.g. by comparing users' game play statistics with those of others, while the level of noise is generally low as game plays require active participation of the user (presuming the user is playing while logged with its own account). The main additional requirement that is imposed to the implicit data is that it should be sampled over a relatively long period in order to acquire an appropriate level of noise reduction. Similar to explicit data, the evaluation itself is of course only related to those games the user actually consumed, neglecting the fact that the actual recommendation never took place. Therefore, real live implementation and evaluation are of essential importance for commercial systems.

The algorithms that are presented in this paper have been extensively tested on a large data set. Based on these results, the most appropriate recommendation algorithm has been implemented on a social gaming network (with minor commercial extensions like e.g. adding newer games to the list, etc.). The selected algorithms are currently under evaluation. This live evaluation requires a long testing period as the performance of the algorithm is evaluated based on the long term user consumption of the game. The description of the results of these live experiments is scheduled as future work.

For the implicit data, the conversion formula as shown in Equation (1), was applied during the experiments for translating the number of game plays into a score in the interval [0.5,1]. 0.5 was chosen as the lower bound of the conversion formula as we can assume that when a game is played at least once by a user, that user has a certain interest in that game, regardless whether he actually liked or disliked that game.

$$r = \frac{1}{2} + \frac{1}{2} \times (arctan(c) \times \frac{2}{\pi})^p \qquad (1)$$

In this formula, c denotes the number of game plays by a specific user for a specific game. Parameter p was calibrated so that 15 game plays of a game resulted in r=0.8 for that game. This implies that a game with 15 game plays by one user, was considered to be found an appealing game by that user. During the experiments 15 game plays was set as a fixed value for each game.

Additionally to the number of game plays, the average game play time of a user was also taken into account during the experiments. Therefore, each user's average game play time for a game was compared with the average game play time of all users for that game. The same Equation (1) was used to calculate this rating but this time with a game dependent variable p, calibrated so that when a user's average game play time for a game equals to all users' average game play time, a score of 0.5 was returned. This score accounted for 50% of a user's total implicit rating for a game. The advantage of taking the average game play time into account, was that users who are good in playing a game or like to play a game for a long time, are awarded with a higher implicit rating for that game.

For the live implementation of the recommender engine, Equation (1) was rescaled to

Equation (2).

$$r = (arctan(c) \times \frac{2}{\pi})^p \quad (2)$$

In this equation, parameter p is changed for each game. Equation (3) shows how p is calculated. G_{AVG} denotes the average number of game plays for that game, while taking the logarithm of 0.5 in the numerator incorporates that the conversion formula results in a score of 0.5 when the user plays a specific game as often as the average number of game plays for that game.

$$p = \frac{log(0.5)}{log(arctan(G_{AVG}) \times \frac{2}{\pi})} \quad (3)$$

Note that for each conversion it should be taken into account that the data set only contains data of a specific sample period. As such, this may lead to impricise conversions. E.g. applying the above conversion formula, an implicit rating starts at, depending on the implicit data interval, e.g. 0 and by increasing the number of game plays this rating will then gradually increase to 1. However, as a user was playing a game very often but only his last game plays for that specific game took place during the data sample period, this will result in a much lower implicit rating for that game than it actually deserves. Similarly, when only the last game play in a set of game plays of a specific game was recorded during the sample period, this may lead to a negative rating while actually the game should have a positive rating. As such, integrating negative feedback on a time-restricted data sample, may entail a specific fault margin which may be propagated both in the training and evaluation data set. Therefore in the lab experiments, a relatively long evaluation period of six months was taken. While imposing extra computational overhead due to the size of this data sample, this greatly reduces the introduced fault tolerance.

4.2 Content-based Filtering

While in this paper, the main focus is on the performance of collaborative filtering techniques on the game data set, two other types of recommendation techniques have been evaluated: a content-based approach and a demographic-based approach. In a content-based approach, item specific characteristics are taken into account in order to recommended new items. As the online gaming platform Gatcha! had already implemented a proper content-based approach for making game recommendations on their site, their algorithm was evaluated on the data set and taken as a reference point for the CF experiments. In this algorithm, the game category was used as the main game property. Typical game categories include sports, action, arcade, puzzle, shooter, etc. Based on the category of the game that was last played by a user, a list of recommended games was generated, composed for 50% by the most popular games (based on the game plays) in the same category and for 50% by the newest games in that same category. If the result set was not large enough, random popular games were added to the result set.

4.3 Collaborative Filtering (CF)

The most common approach to CF is based on neighborhood models (k-NN) where a user-item preference matrix is interpolated from ratings of similar users and/or items. The user approach is based on predictions of ratings of the same item by similar users [22] while the item neighborhood approach identifies pairs of items that tend to be rated similarly, in order to predict ratings for an unrated item based on similar items by the same user [23]. The item based approach generally [23] tends to offer better scalability and a higher accuracy than the user based approach. Despite these advantages of item based over user based CF, both approaches were evaluated on the test data be-

cause of the atypical characteristics of the data.

During the experiments, several similarity metrics have been evaluated. These metrics are used to indicate the similarity between two items.

4.3.1 Euclidean Distance

One of the most common (and simplistic) metrics is the Euclidean distance metric. This metric describes the distance between two points in a two-dimensional plane. This formula is shown in Equation (4). In this formula, X_i is the rating giving by user X to game i. In order to reduce the computational complexity, the square root in the denominator was removed from the formula. This can be done as these metrics are only applied to calculate the mutual order of the different games. The Euclidean distance returns a value in the interval [0,1], where 1 indicates a maximum similarity.

$$s_e(X,Y) = \frac{1}{1+\sum_{i=1}^{n}(X_i - Y_i)^2} \tag{4}$$

4.3.2 Pearson Correlation

Another popular similarity metric is the Pearson correlation coefficient. This coefficient takes the rating behavior of different users into account, as some users tend to give consistently higher or lower scores to items than other users. When the absolute difference between the ratings of both users is also consistent, the Pearson correlation coefficient will indicate there is a high similarity between both users. The Pearson correlation coefficient is shown in Equation (5).

$$s_p(X,Y) = \frac{\sum_i (X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_i (X_i - \overline{X})^2 \sum_i (Y_i - \overline{Y})^2}} \tag{5}$$

$\overline{X}$ is the average rating of that user over all games that user has rated. This metric is only calculated when users have at least rated two similar games. It returns a value in the interval [-1,1]. A value of 1 implies that a linear equation describes the relationship between both users perfectly. A value of 0 implies that there is no linear correlation between the users.

4.3.3 Cosine Similarity

Cosine similarity is a measure of similarity between two vectors in the m dimensional user-space, by measuring the cosine of the angle between them. The result of the Cosine function is equal to 1 when the angle is 0, and it is less than 1 when the angle is of any other value.

$$s_c(X,Y) = \frac{X.Y}{||X||\,||Y||} = \frac{\sum_i X_i Y_i}{\sqrt{\sum_i (X_i)^2}\sqrt{\sum_i (Y_i)^2}} \tag{6}$$

Similar to the Pearson correlation coefficient, cosine-based similarity takes the different rating or playing behavior of users into account (achieved by the denominator in Equation (6). Where Pearson handles absolute differences in ratings or number of game plays, cosine similarity handles the proportional differences in ratings or game play numbers.

4.3.4 Jaccard Index

The fourth similarity metric that was evaluated, is the Jaccard index or Jaccard similarity coefficient, shown in Equation (7).

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} \tag{7}$$

This index measures similarity between different data sets, and is defined as the size of the intersection divided by the size of the union of the data sets, in this case the games played by a user. The Jaccard coefficient returns a value in the interval [0,1].

4.4 Evaluation Methodology

Within the field of recommendation systems, several evaluation metrics have been defined. One of the most commonly used metrics is the Mean Absolute Error (MAE), a quantity used to measure how close predictions are to the evaluation data. A different kind of evaluation metrics focuses on the fact how good a recommendation engine is able to distinguish correct from incorrect predictions. Receiver Operating Characteristic (ROC) curve analysis is a well-known example of this type of evaluation metric. A ROC curve is drawn by plotting the sensitivity, or *true positive* rate, versus the *false positive* rate (1 - specificity or 1 - *true negative* rate), for a binary classifier system while its discrimination threshold is varied. For an ideal classifier, a ROC curve starts in the origin, then goes straight up to the point (0,1) and then ends in (1,1). In data mining, the Area Under Curve (AUC) is often taking into account. The AUC is a value between 0 and 1 and is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. The higher the AUC, the better the recommendation engine.

For both the content-based and collaborative filtering algorithms, different use case scenarios were evaluated. Each use case scenario was determined by several configuration settings, determining how and which data should be taken into account. In order to reduce the execution time of these different experiments, all experiments were split up into several subprocesses that were executed in parallel on several nodes of the High Performance Cluster (HPC) of Ghent University. Those settings that had most impact on the evaluation results were:

- Type of data: explicit, implicit or both.
- userCO, the required number of ratings a user should have in order to be taken into the training or evaluation set. Ranged from 1 to 60.
- topN, the N number of top recommendations that are taken into account, e.g. topN=5 only evaluates the 5 games that received the highest rating by the recommendation system. Ranged from 1 to 20.
- rocCO, the limit that determines whether ratings should be considered as good or bad. Naturally, this value is within the range [0,1].

For the evaluation of the content-based approach only ROC curve analysis was applied as this algorithm cannot predict ratings, required to calculate the MAE. The ROC curve was drawn by taking into account explicit and/or implicit ratings of 1000 users with an extensive profile (i.e. they had at least rated 20 different games). For every game a list with related games was first composed, according to the Gatcha! algorithm. For each user, a recommendation list was then built composed of all the recommendations that would be made in case each game he or she had played or rated was taken as a start scenario. For each user, each game in his explicit or implicit data set was compared with the compound recommendation list of that user. When the played or rated game was found in the recommendation list, that item was added to the ROC curve dataset as (1,*rating*), with *rating* the user's explicit rating or converted implicit rating for that game. When the game was not found in the recommendation list, (0,*rating*) was added to the ROC curve dataset. This methodology allowed us to evaluate whether recommended items (which are assumed by the recommender system as being positive) actually had a positive rating for that user.

For the evaluation of the k-NN based CF algorithms, both the ROC curve analysis and MAE calculation were taken into account.

MAE calculation could be applied as CF algorithms predict a rating for each game. Taking this metric into account allowed us to make a more profound analysis of these experiments. For these k-NN based CF experiments, the data set was randomly split into 80% training data and 20% test data.

5 Evaluation Results

5.1 Content-based Recommendation

The evaluation of the content-based algorithm as deployed at Gatcha! was evaluated for 1000 users with a relatively extensive profile. Although several scenarios, taking into account explicit data or implicit data or a combination thereof, were evaluated all scenarios yielded very similar ROC curves. One of these curves is shown in Figure 2, where the evaluation was performed on explicit data, and only the best 5 recommendations were taken into account.

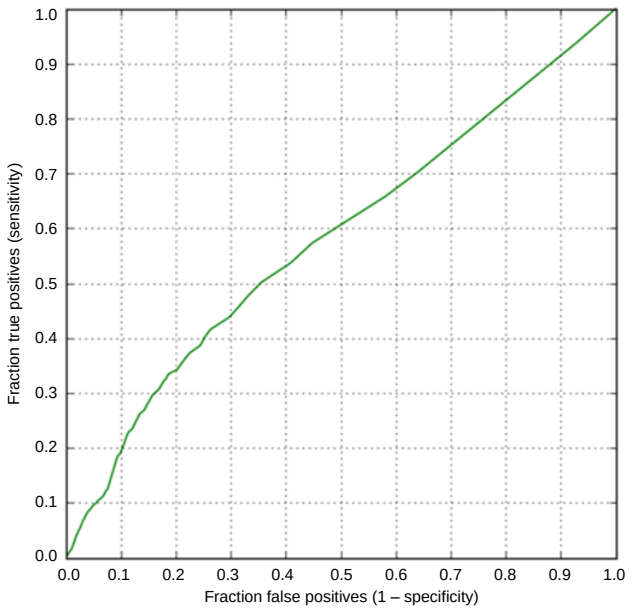


Fig. 2. ROC curve content-based CF on the explicit dataset with userCO=20 and topN=5.

As shown in Figure 2, this scenario resulted in a very low Area Under Curve value AUC=0.580, however this was still the highest AUC value that was found during all these experiments. The results learned that this type of recommendation hardly performed better than a random recommendation engine. This low AUC value can be explained by the gaming behavior of many gamers where the consumption of one type of games (over a relatively long period) often results in a reduced interest in similar games. This can be illustrated by an example: while in many products or movie recommender systems, very similar items, e.g. The Lords of The Rings movies, have very similar ratings, an online social gamer may often not consume or dislike games which are very similar to those he is/has been playing. E.g. when an online social gamer is playing a strategic game then he might not be interested to play other strategic games during the same period. Similarly, when a user has played a game like e.g. Farmville, then even when he quit playing the game after some time, chances are low he or she starts developing a new farm in one of the many Farmville clones. It can be concluded that the Content based approach hardly performs better than a random classifier, indicated in a ROC-curve by a diagonal curve.

5.2 User-based Collaborative Filtering

User-based collaborative filtering makes recommendations for a specific user based on the behavior of those users that have the most similar profile. A major drawback [23] of user-based CF is that it is not very suitable for large datasets as this makes the calculation process very time consuming. As such, with the huge number of users the Gatcha! data sample consists of, the experiments, even when executed in parallel on several CPU nodes, were very time-consuming (as reported on below).

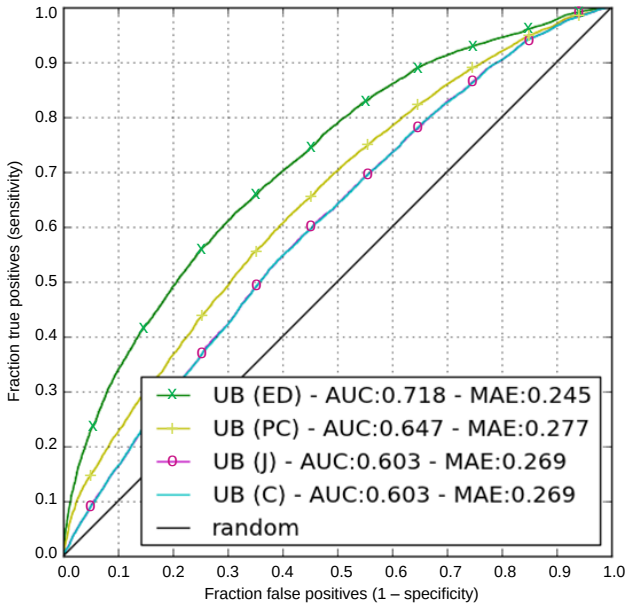


Fig. 3. ROC curve and MAE for user-based CF on the explicit dataset with userCO=20 and topN=5.

Figure 3 shows the ROC curve together with the calculated Area Under Curve (AUC) and Mean Absolute Error (MAE) values for an experiment on the explicit data set, evaluating four different similarity metrics: Euclidean distance (ED), Pearson Correlation (PC), Jaccard index (J) and cosine similarity (C). The calculations were restricted to only those users who have at least rated 20 different games (userCO=20), and the evaluation was executed by taking into account the 5 games which received the highest rating by the algorithm (topN=5) for each user. As the figure shows, the Euclidean Distance was the best similarity metric with an AUC=0.718. However, the MAE for each similarity metric was quite high as even Euclidean distance (ED) had an average MAE=0.245 or 3.205 on a scale of 10. The execution time of the experiment, even with userCO=20, was around 24 hours. Similar (or worse) results were found during other user-based experiments. The relatively high MAE, together with the moderate AUC and high execution time made user-based CF not suitable for this use case scenario.

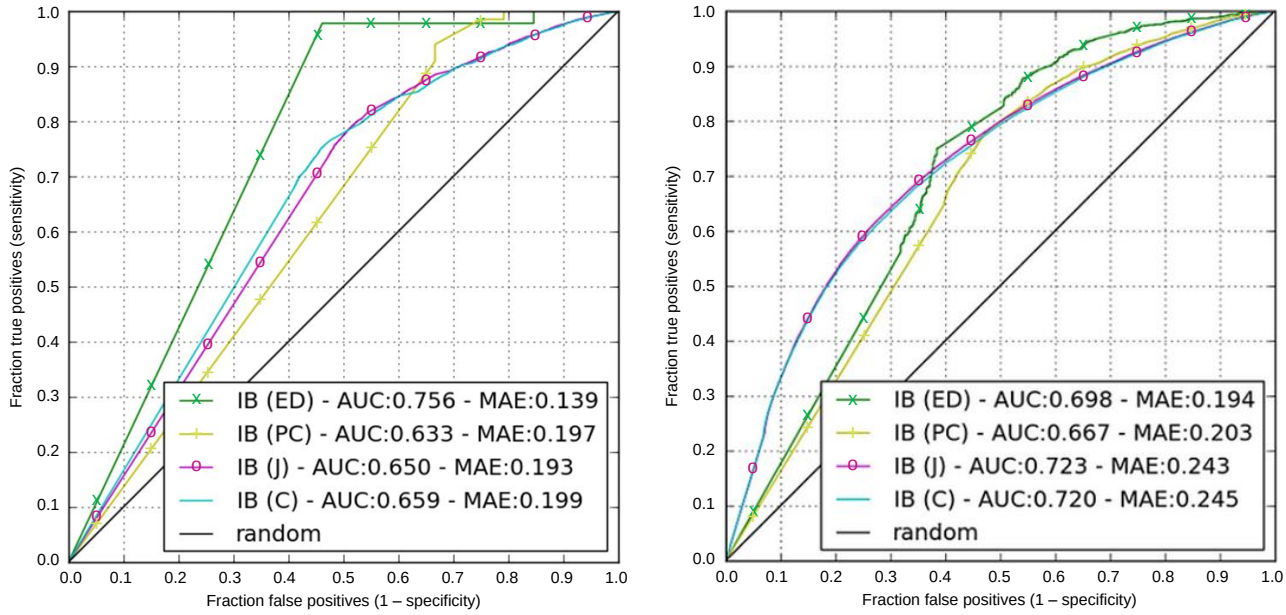
In the experiment shown in Figure 3, ratings starting from 9, which corresponds to a rocCO=0.87, were considered as positive ratings. During all experiments, both user and item based, it was found that for high values for rocCO, the limit between positive and negative ratings, resulted in the best experiment results when explicit data was taken into account. Specifically when rocCO corresponds to 8 or 9 star ratings, the highest AUC and MAE values were found. These high values are a consequence of the extreme rating behavior of users, as previously mentioned in section 3.1. Similarly, for implicit ratings a userCo=0.87 was found to produce the best results. For consistency reasons, all experiment results that are described in this paper have the same rocCO=0.87 configuration.

5.3 Item-based Collaborative Filtering

5.3.1 Explicit Ratings

Item-based CF recommends items based on the similarity between those items. In the explicit use case scenario, the similarity between games is based on the ratings users gave to those games. Based on the games that users have rated and their similarity with other games, new games are then recommended to the users. As the item similarity (matrix) can be calculated in advance and only has to be calculated once for all users, item-based CF is far less time consuming than user-based CF. For the explicit ratings scenario most experiments required less than an hour execution time on the HPC, even when taking more than one million users into account. The evaluation results of two experiments with fixed userCO=5 and variable topN configuration are shown in Figure 4. In Figure 4(a) the top 5 recommendations for each of those users were evaluated (topN=5) while in Figure 4(b) topN was set to a relatively high topN=60.

Both figures show that the Euclidean dis-



(a) topN=5 (b) topN=60

Fig. 4. ROC curve and MAE for item-based CF on explicit data with userCO=5.

tance (ED) similarity metric clearly makes better recommendations than the Pearson correlation (PC). This was the case in almost all item-based explicit data experiments. This is quite remarkable as in most use cases, Pearson correlation is supposed to outperform the Euclidean distance. This can be explained by the extreme rating behavior of gamers as they all tend to give extreme high or low ratings, negating the advantage of Pearson correlation that takes different rating patterns into account. Additionally, as most users only tend to positively award games, with 10 star ratings having by far the greatest share in the provided explicit data set, the average star rating of each user was often so high that the effect of all ratings lower than this high average (even when an 9 star rating was given) immediately were negatively enlarged.

While Euclidean distance clearly outperforms all other similarity metrics for lower topN values in terms of AUC, with an AUC of up to 0.769, which is in general considered to be fairly good, Figure 4(b) shows that for higher topN values its AUC value was decreasing while those of Jaccard index (J) and Cosine similarity (C) were increasing up to 0.804 for the scenario were topN=60 and userCO=20. With respect to ROC curve analysis, the AUC values for Pearson correlation results were found to be ranging most of the time between 0.615 and 0.657, which corresponds to a rather weak classifier. Only in the scenario were topN=60, Pearson was performing remarkably better for all experiments with a maximum AUC of 0.752 for userCO=20. Note that the straight lines at the beginning of the ROC curves are due to the monotype rating of users. With many profiles only containing 10 star ratings, many recommendations predict 10 star ratings (or score 1.0) for other games. While drawing the ROC curves, these 1.0 ratings are all classified as false or true at the same time which results in these long straight curves.

Table 3. AUC and MAE values for item-based CF on explicit data with topN=20 and variable userCO [*uCO*].

	uCO	**ED**	**PC**	**J**	**C**
AUC	5	0.746	0.657	0.695	0.690
	10	0.731	0.626	0.729	0.724
	20	0.707	0.640	0.774	0.769
MAE	5	0.167	0.213	0.225	0.230
	10	0.151	0.185	0.217	0.221
	20	0.149	0.205	0.196	0.200

Table 3 shows the AUC and MAE values for item-based CF on explicit data with topN=20 and variable userCO. As shown in this table, increasing userCO had a beneficial effect on the AUC results of both Jaccard and Cosine similarity. For the Euclidean distance and Pearson correlation AUC results were varying depending on the userCO configuration but without any clear pattern.

With respect to the MAE values, an increasing userCO had a beneficial effect on all four metrics during almost all experiments, while an increasing topN had the opposite effect for all four metrics. During all experiments, the Euclidean distance achieved by far the best MAE scores with values ranging between 0.125 and 0.194. These values correspond with a star value of respectively 2.1 and 2.75. These values are remarkable better than those of the user-based CF experiments.

5.3.2 Implicit Ratings

In the implicit use case scenario, the game recommendations and similarity calculation between games is based on the game plays of the users for each game. As the implicit data set contained much more data than the explicit data set, most experiments required several hours of execution time on the HPC. This time was mainly used for creating the item similarity matrix as all game plays of a period of six months were taken into account during the experiments. The reason for this long training period was explained in section 3.2. Table 4 shows the AUC and MAE values for item-based CF on implicit data with topN=20 and variable userCO. These are the exact same configuration settings as Table 3 from Section 5.3.1 regarding the explicit ratings evaluation results.

Table 4. AUC and MAE values for item-based CF on implicit data with topN=20 and variable userCO [*uCO*].

	uCO	**ED**	**PC**	**J**	**C**
AUC	5	0.769	0.722	0.689	0.678
	10	0.759	0.700	0.667	0.659
	20	0.689	0.690	0.670	0.654
MAE	5	0.127	0.132	0.099	0.100
	10	0.124	0.140	0.102	0.102
	20	0.159	0.143	0.102	0.102

As shown in Table 4, Pearson correlation performs slightly better than Jaccard index and Cosine similarity in terms of AUC values while the opposite remark can be made when the MAE evaluation is taken into account. Contrary to the explicit data scenario, the configuration of parameter userCO has little impact on both the AUC and MAE performance of these metrics. While Euclidean distance is performing better than the other three similarity metrics for lower userCO values, this similarity metric was not producing reliable results for most experiments, especially for lower topN value where ROC curve analysis for Euclidean distance obtained rather extreme values, often approaching a perfect classifier. Analysis of these results, learned that appliance of Euclidean distance always returned only a very limited list of recommendations. This is due to

the extreme popularity of a limited set of games which have a very low similarity with other games. As most (new) players often start with playing only those popular games the low similarity with those other games resulted in low estimated ratings for those less popular games. The application of equation (3) may improve this disadvantage.

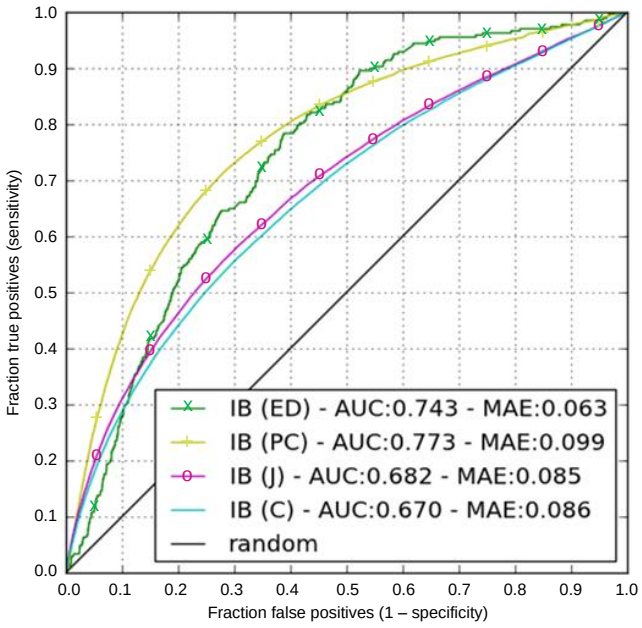


Fig. 5. ROC curve and MAE for item-based CF on implicit data with userCO=5 and topN=60.

When comparing the results of the explicit experiments in Table 3 with those of the implicit experiments of Table 4 in terms of ROC curve analysis, Jaccard index and Cosine similarity are performing worse in the implicit scenario while for Pearson correlation the opposite observation can be made. However, when comparing the MAE values of this implicit scenario with those of its explicit counterpart, significant lower MAE values were found. Only Euclidean distance was sometimes performing slightly worse. Lower MAE values incorporate better predictions of the estimated number of game plays. In all implicit scenarios, for both high and low topN and userCO configurations, significant lower MAE values were obtained than when using explicit data, even for Euclidean distance. The main reason for these lower MAE values is that for most users relatively more implicit ratings than explicit ratings are known, resulting in more detailed predictions for these users. These conclusions are also shown when comparing Figure 5 with Figure 4(b) from the explicit data Section 5.3.1.

In the experiment from Figure 5, we evaluated almost all predictions that were made by taking into account the top 60 predictions for each user (topN=60). Similar to the results that were found in Table 4, Pearson correlation is performing better than Jaccard and Consine in terms of ROC curve analysis with an AUC=0.773, which approximates a good classifier. This trend was found in all implicit rating scenarios, for both high and low topN and rocCO configurations. When comparing these results with those of Table 4, it is shown that the configuration parameter topN has a positive impact on the AUC performance of Pearson correlation for increasing topN values. With respect to the MAE evaluation, an increasing topN configuration was found to have a positive impact on all four metrics.

5.3.3 Combined Ratings

The last series of item-based experiments evaluated the impact of combining the explicit and implicit ratings. To combine both types

of ratings, the absolute values of the predicted ratings were taken into account for each item. A specific ratio was applied between those absolute values. Although there is no exactly quantified relationship between the absolute explicit and implicit ratings, combining both types of ratings may result in a new type of ratings that after evaluation may prove, by means of a selected classifier, to perform well for specific use case scenarios or by applying specific configuration parameters. Multiple ratios for combining both types of ratings were evaluated. When no explicit data for a user was found, only the implicit data was taken into account. After evaluating several scenarios it was concluded that an equal combination of both types of data yielded the best results. Only for Euclidean distance a higher impact ratio for explicit data resulted in higher AUC values. Especially Pearson correlation was found to produce very stable and relatively high AUC values in all scenarios, independent of the mutual ratio which was applied. Figure 6 shows the results of an experiment where both the explicit and implicit results were equally taken into account.

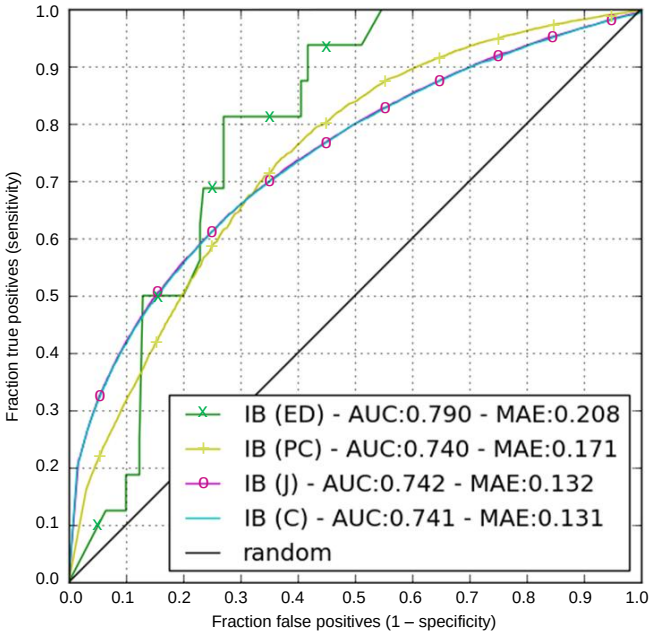


Fig. 6. ROC curve and MAE for combined item-based CF with a 1-1 ratio for the implicit and explicit ratings. userCO=10 and topN=10.

Figure 6 shows that while Jaccard index and cosine similarity performed approximately equally in terms of AUC values, Euclidean distance and Pearson correlation performed slightly better. Especially for low topN values, Pearson correlation produced better results than the other metrics with an AUC up to 0.804 while the weak performance of Euclidean distance with respect to implicit data made this metric less suitable for lower topN values. In terms of MAE values, Jaccard and cosine performed in all scenarios very similar, producing slightly better results than Pearson correlation.

In general, it can be stated that taking into account both explicit and implicit data yielded good results for both the AUC and MAE metrics in all scenarios, while only slightly performing worse compared to the strengths of these two other approaches in very specific scenarios. With respect to those combined ratings, Pearson correlation was found to be the most interesting metric as its appliance resulted in the highest AUC values, making it a good classifier between good and bad recommendations

while its MAE value, although slightly higher than the other metrics, is still considered to be relatively low. It should be noted that in a real use case scenario only a few recommendations are made. This corresponds to lower topN values which makes Pearson the most appropriate candidate for the actual implementation.

6 Conclusions

In this paper, we have evaluated the applicability of neighborhood-based collaborative filtering (CF) algorithms for the recommendation of online social games. These evaluations were based on a large data set of an online social gaming platform containing game ratings (explicit data) and online gaming behavior (implicit data) of millions of active users. An existing content-based recommendation algorithm was first evaluated in order to have a valid reference. It is shown that neighborhood-based CF algorithms greatly outperform the existing content-based approach. Of these neigborhood-based CF algorithms, the item-based approach was by far the most appropriate technique surpassing the user-based approach not only based on the evaluation results but also by its greater applicability as the user-based approach showed scalability issues while dealing with large amounts of data.

The performance of the item-based approach was further evaluated for several different scenarios. Each scenario was determined by multiple configuration parameters of which the number of top recommendations and the number of ratings per users were found to be the most important. Different similarity metrics were taken into account and experiments were run on both explicit and implicit data or a combination thereof. During the explicit data experiments, Euclidean distance outperforms the other metrics with respect to the estimation of ratings, obtaining lower MAE values for all scenarios. Additionally, for the scenarios were only a small number of the top N recommendations were evaluated, Euclidean distance obtains the highest AUC scores, corresponding to a reasonable good recommendation system. These rather unusual results can be explained by the extreme rating behavior of online gamers as described in the paper. When more top N recommendations were taken into account, Jaccard index and cosine similarity obtain the highest AUC values. When using the implicit data set, all similarity metrics have lower MAE values during all scenarios when compared to the same scenarios when using explicit data. While the Euclidean distance has difficulties coping with the fact that only a limited amount of games is extremely popular, Pearson correlation acquires the highest AUC ratings, which in some cases was even better than during the explicit scenarios. However, in general the AUC values of the implicit scenarios are found to be lower than those of the explicit scenarios. Finally, the combination of both explicit and implicit results was evaluated. These results show that a combined approach yields overall good results on all evaluation metrics for all scenarios, while only slightly performing worse compared to the strengths of the two other approaches. Pearson correlation obtains the highest AUC values, especially for low top N configurations, while its MAE value, although slightly higher than the other metrics, are still considered to be relatively low. As in a real use case scenario only a few recommendations are shown on the website, combined item-based CF with Pearson correlation as the distance metric was finally chosen as the most appropriate candidate for the actual implementation. This algorithm has been implemented on the Gatcha! platform and is currently running on a platform that is daily used by the entire Gatcha! gamer community.

Research has shown [24] that taking into account time information, i.e. when an item was released and purchased or in our scenario

a game was played, may further improve the accuracy of the collaborative filtering based recommender systems. We are currently extending our research on the recommendation of online social games by taking the game play dates into account for the conversion of gaming data to an implicit rating. One of the objectives of these experiments is to find the most appropriate time degradation and game play conversion functions and to evaluate their impact on the overall performance of the recommendation engine.

Acknowledgment The authors would like to thank Pieter De Schepper and Toon Coppens from Massive Media and Jan Heuninck (former HoGent student) for the efficient and pleasant cooperation and their valuable contributions to the work presented in this paper.

References

[1] Nielsen. Nielsen NetView: June 2009-June 2010. 2010.

[2] GP Bullhound. Social Gaming: the fastest growing segment of the games market. 2010.

[3] InfoSolutionsGroup. PopCap Social Gaming Research Results. 2010.

[4] Gatcha!. http://netlog.com/play. 2012

[5] Netlog. http://netlog.com/go/about. 2012

[6] Konstan J A, Miller B N, Maltz D, Herlocker J L, Gordon L R, Riedl J. GroupLens: applying collaborative filtering to Usenet news. *Communications of the ACM*, 1997, 40(3):77–87.

[7] Bell R M, Koren Y. Lessons from the Netflix prize challenge. *SIGKDD Explorations Newsletter*, 2007, 9(2):75–79.

[8] Leung C, Chan S, Chung F-l, Ngai G. A probabilistic rating inference framework for mining user preferences from reviews. *World Wide Web*, 2011, 14(2):187–215.

[9] Go G, Yang J, Park H, Han S. Using Online Media Sharing Behavior as Implicit Feedback for Collaborative Filtering. In *Proc. of SocialCom*, Sept., 2010, pp. 439–445.

[10] Shen D, Sun J-T, Yang Q, Chen Z. A Comparison of Implicit and Explicit Links for Web Page Classification. In *Proc. of WWW*, May, 2006, pp. 643–650.

[11] Pessiot J-F, Truong T-V, Usunier N, Amini M-R, Gallinari P. Learning to Rank for Collaborative Filtering. In *Proc. of ICEIS (2)*, June, 2007, pp. 145–151.

[12] Liu N N, Yang Q. EigenRank: a ranking-oriented approach to collaborative filtering. In *Proc. of SIGIR*, July, 2008, pp. 83–90.

[13] Liu N N, Xiang E W, Zhao M, Yang Q. Unifying explicit and implicit feedback for collaborative filtering. In *Proc. of CIKM*, Oct., 2010, pp. 1445–1448.

[14] Golbeck J. Generating Predictive Movie Recommendations from Trust in Social Networks. In *Proc. of iTrust*, May, 2006, pp. 93–104.

[15] DuBois T, Golbeck J, Kleint J, Srinivasan A. Improving Recommendation Accuracy by Clustering Social Networks with Trust. In *Proc. of RecSys Workshop on Recommender Systems and the Social Web*, Oct., 2009.

[16] Li H-Q, Xia F, Zeng D, Wang F-Y, Mao W-J. Exploring Social Annotations with

the Application to Web Page Recommendation. *Journal of Computer Science and Technology*, 2009, 24(6):1028–1034.

[17] Wong K W, Zhou S, Yang Q, Yeung J M S. Mining Customer Value: From Association Rules to Direct Marketing. *Data Mining and Knowledge Discovery*, 2005, 11(1):1384–5810.

[18] Drachen A, Canossa A. Towards gameplay analysis via gameplay metrics. In *Proc. of MindTrek*, Oct., 2009, pp. 202–209.

[19] Medler B, John M, Lane J. Data cracker: developing a visual game analytic tool for analyzing online gameplay. In *Proc. of CHI*, May, 2011, pp. 2365–2374.

[20] Lee W S. Collaborative Learning for Recommender Systems. In *Proc. of ICML*, June, 2001, pp. 314–321.

[21] Hu Y, Koren Y, Volinsky C. Collaborative Filtering for Implicit Feedback Datasets. In *Proc. of ICDM*, Dec., 2008, pp. 263–272.

[22] Herlocker J L, Konstan J A, Borchers A, Riedl J. An algorithmic framework for performing collaborative filtering. In *Proc. of SIGIR*, Aug., 1999, pp. 230–237.

[23] Sarwar B, Karypis G, Konstan J, Reidl J. Item-based collaborative filtering recommendation algorithms. In *Proc. of WWW*, May, 2001, pp. 285–295.

[24] Lee T Q, Park Y, Park Y-T. A time-based approach to effective recommender systems using implicit feedback. *Expert Systems with Applications*, 2008, 34(4):3055–3062.