Implicit Feedback Techniques on Recommender Systems applied to Electronic Books.

Edward Rolando Nuñez Valdez; Juan Manuel Cueva Lovelle; Oscar Sanjuán Martínez; Vicente García Díaz; Patricia Ordoñez De Pablos; Carlos Enrique Montenegro Marín. Implicit Feedback Techniques on Recommender Systems applied to Electronic Books. Computers in Human Behavior. 28 - Issue 4, pp. 1186--1193. DOI: https://doi.org/10.1016/j.chb.2012.02.001

Implicit Feedback Techniques on Recommender Systems applied to Electronic Books.

Edward Rolando Núñez -Valdéz[a]*, Juan Manuel Cueva Lovelle[a], Oscar Sanjuán Martínez[a], Vicente García-Díaz[a], Patricia Ordoñez de Pablos[a], Carlos Enrique Montenegro Marín[b]

[a] *University of Oviedo, Department of Computer Science, Sciences Building, C/Calvo Sotelo s/n 33007, Oviedo, Asturias, Spain.*

[a] *nunezedward@uniovi.es, cueva@uniovi.es, sanjuan@uniovi.es, garciavicente@uniovi.es, patriop@uniovi.es.*

[b] *Distrital University, Engineering Faculty, Bogotá (Col),* cemontenegrom@udistrital.edu.co.

* Corresponding author

Abtract: The goal of this research is to define and capture a series of parameters that allowed us to perform a comparative analysis and find correlations between explicit and implicit feedback on recommender systems. Most of these systems require explicit actions from the users, such as rating and commenting. In the context of electronic books this interaction may alter the patterns of reading and understanding of the users, as they are asked to stop reading and rate the content. By simulating the behavior of an electronic book reader we have improved the feedback process, by implicitly capturing, measuring, and classifying the information needed to discover user interests. In these times of information overload, we can now develop recommender systems that are mostly based on the user's behavior, by relying on the obtained results.

Keywords: Recommender system, eBook, implicit feedback, explicit feedback.

1 INTRODUCTION

Due to the large amount of information available on the Internet, sometimes it is difficult for users to find the content that they really need in a quick and easy way. The user tends to: seek for recommendations from others who have previously had the same needs; or select those items that are closest to what they were looking for [1].

The use of recommender system as an information retrieval technique attempts to solve the problem of data overload. They filter the information available on the web and help users to find more interesting and valuable information [2-4].

In order for recommender systems to be more effective we believe that is necessary to enhance the feedback process. We need to implicitly gather as much information related to the user profile as possible, so to be able to measure the user's interest about an item or group of items. As illustrated in [5], the most common solutions and the more prevalent are the ones based on explicit ratings. These techniques can alter the user's regular navigation and reading patterns, because they have to stop and rate the items.

By defining a collection of implicit parameters, comparatively analyzing their values, and measuring their correlations, we infer the grade of interest that users may have for certain items while interacting with an electronic book. This process allows us to convert implicit values into explicit ratings that help the recommender system make more precise recommendations. The remainder of this paper is structured as follows: in section 2 we describe the main problems with existing recommender system in electronic books; in section 3 we present the state of art of recommender systems; section 4 shows our case study; and finally, in section 5 we explain our conclusions and possible future work.

2 PROBLEMS

To efficiently capture and measure the interaction parameters between a user and an electronic book, and implement a recommender system suitable for these types of devices, we must take into consideration a number of problems. In general, we can say that there are three major problems associated with this subject [6].

2.1 Information overload

The access to tremendous amount of data available on the Internet requires mechanisms and classification algorithms to optimize the search of information and access these contents efficiently. The amount of information available on the Web increases every day, and this becomes an optimization problem for recommender systems [2, 7, 8].

2.2 Implementation of an efficient feedback mechanism

In most cases, feedback mechanisms are based on explicit feedback, and this may cause inconvenience to the users, as they typically do not like rating contents.

Explicit ratings are the most common and obvious indicators of the user's interest, because it allow them to tell the system what they really think of the rateable objects. On the other hand, they alter the user's regular navigation and reading patterns, because they have to stop and rate the items. In addition, the users may not rate the objects if they do not perceive any benefit [5].

Therefore, we believe it is necessary to capture as much information as possible without the direct intervention of the users, in order to [1] determine their interests and needs and try to implement a more effective feedback mechanism.

2.3 Limited computing capability in electronic book devices

The memory and CPU consumption of any recommender system is very high as they have to deal with lots of data. The algorithms optimization to improve its performance is one of the main fields of research in this area.

A constant characteristic of these systems is the processing of constantly altered data (real time), which requires efficient algorithms with a low cost of execution.

A recommender system requires a continuous learning about the user's profiles and a constant update of the system's information. And so, it is necessary to minimize memory and CPU usage during the feedback retrieval.

As eBooks have certain limitations of computation and storage, it is necessary to evaluate and design a methodology that enables these devices to update and store the object's ratings.

This would allow recommender systems to operate effectively and without relying on external technologies on an ongoing basis. We need a synchronization mechanism of the data available on external servers. This can be implemented either through Web services or through a synchronization process against a desktop computer application. This synchronization must end up with all the user profile's information being stored in the electronic device using a standard format.

3 STATE OF ART OF RECOMMENDER SYSTEMS

Today recommender systems are very useful on the Web and are widely used, these help users find content that is interesting to them easily, quickly and without much effort. These contents are selected by recommender systems of a large amount of content that are available on the web.

In general, a recommender system is defined by [9] as *"A system that has as its main task, choosing certain objects that meet the requirements of users, where each of these objects are stored in a computer system and characterized by a set of attributes."*

These consist of a series of mechanisms and techniques applied to the retrieval of information to try to resolve the problem of data overload on the Internet. These help users to choose the objects that can be useful and interesting, these objects can be any type, such as books, movies, songs, websites, blogs [8].

Recommender systems are based on personalized information filtering, used to predict whether a particular user likes a particular item (prediction problem), or identify a set of N items that may be of interest to certain users (*top-N* recommendation problem) [10].

3.1 Recommender system classification

Recommender systems can be classified into different types according to the type of information that used to make recommendations [11, 12].

Traditionally there are several paradigms of filtering information used to generate recommendations, these are classified as:

- **Content-based:** these try recommend similar contents to another that liked to a particular user in the past.
- **Collaborative filtering** identifies users whose tastes are similar to a given user and recommends to this user the contents that likes to the other users.
- **Hybrid approach:** is a combination between between content-based and collaborative filtering.

Other variety of techniques have been proposed for performing recommendation by other authors as [12], although one way or another, these are related with the classifications of recommender systems mentioned above, these include: Demographic recommendation, Knowledge based recommendation, Utility based recommendation.

Currently there are a wide range of recommendation systems that are used in different areas, whether for commercial or scientific or experimental purposes. For example: PHOAKS [13], Referral Web [14], Fab: content-based collaborative recommendation [15], Amazon.com recommendations: item-to-item collaborative filtering [16].

3.2 Feedback techniques

The recommender systems collect user information through the feedback techniques, and stored in users profile in order later to reflect your interests and make recommendations. The feedback techniques are classified into two types: Explicit and Implicit feedback [10, 11, 17].

The combination between explicit and implicit feedback techniques provides another paradigm for recommender systems, despite that these exhibit different characteristics about users' preferences [18].

3.2.1 Explicit feedback

Through a survey process, the user evaluates the system by assigning a score to an individual object or a set of objects. Explicit feedback provides users with a mechanism to unequivocally express their interests in objects [18].

Figure 1 shows the most common explicit feedback system used by users on the web to express their interest by objects.

Figure 1: Most common explicit feedback systems.

For example, Amazon online store, Film affinity, Movilens and other, use the *star ratings system* that allows users to indicate which products are of their interest.

On the other hand, social networks as Facebook, YouTube and other use the *Like rating system* to allow the users to rate the contents.

Finally, Google+1 is a new feature that Google added to its search engine so users can evaluate explicitly the websites that like them. So, they recommends website to their contacts.

3.2.2 Implicit feedback

This process consist in evaluate the objects without interventions of users. Namely, this evaluation is performed without the user being aware, through capture of information obtained from the actions made by the users in the application. For example, when the user accesses to a news or read an article online, according to the time it takes for reading, the system could automatically infer whether the content is on his interest.

Implicit feedback techniques have been used to retrieve, filter and recommend a variety of items: movies, journal articles, Web documents, online news articles, books, television programs, and others. These techniques take advantage of user behavior to understand user interests and preferences [19].

Types of implicit feedback include purchase history, browsing history, search patterns, or even mouse movements. For example, a user that purchased many books by the same author probably likes that author [20].

4 CASE OF STUDY

To achieve an approach to the solution of the explicit feedback, we developed an application based on **eInkPlusPlus** project, and contain a series of photo books sorted by categories. Each category and photo book is composed by the same amount of objects. Specifically, each category contains 10 photo books and each photo book contains 10 pictures, this is so that each object has the same probability assessment. We choose photo books because we think that the interaction with them is more comfortable, fast and efficient than the complete e-books reading. This enables the users to navigate through several photo books in the shortest time possible, allowing us to extend the tests to a greater number of users. The application is designed like a library books that consists in:

- **Categories:** Categories represent the classifications of books (e.g., comics, computer and internet, novels, biographies, science, etc.).

- **Photo books:** Each photo book represents a reading object (e.g., a book, a magazine, a scientific paper, etc.). From now on we will call it "content".

- **Photos:** Each photo is a page of a content, which users can view and interact with it, allowing the user to forward or back one page to another. From now on we will call it "items".

The users that interact with the application can browse the different categories, contents and items. Each user can view individual items of the contents, comment the contents, send these to his friends and explicitly assess these, indicating which are of his interest.

On the other hand, transparently to users, we recorded the user's interaction with each object (category, content and item) of the application, to capture the implicit parameters and determine the number of times a user visits a category, content or item, the time taken per session reading it, etc.

This application has been distributed to 58 users with different skill levels, different ages, without prior knowledge of the contents and selected at random, which provided the data necessary to carry out the study said.

Later we will describe how the data were obtained and the relations established between them. Subsequently, an analysis of the same and will present final conclusions.

4.1 Graphic User Interface

The Graphical User Interface is a ubiquitous web application developed in **RubyOnRails** and can be run on any device with a Web browser (e.g., Mozilla Firefox, Microsoft Internet Explorer, Google Chrome, etc.). In this Web application we can register as a user, create contents, add items to the contents, to comment the contents, browse the different options of the application, etc.

As shown in Figure 2, when a registered user is logging the application shows the homepage with different categories, through which the user can navigate and access different content.

Each category shows the contents that belong to it, including the content cover image, title and author of contents. Clicking on the title or on the cover the users access the selected content.

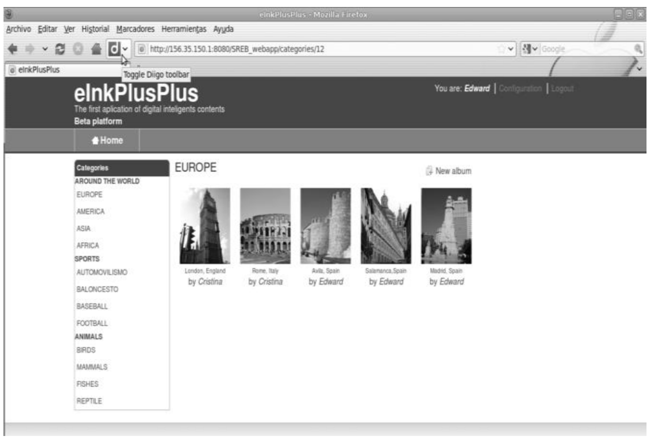

Figure 2: Graphical User Interface**.**

4.2 Catching explicit parameters

To perform the analysis and comparison between implicit and explicit feedback, we need some way to know the real value of the user with respect to content (explicit evaluation).

The best way to know if a user is interesting in content is through explicit feedback, so as shown in Figure 3, in this study defined several explicit feedback systems.

Figure 3: Photo books viewer.

4.2.1 Content rating system

In this system there are five stars by which the user can rate the contents. Score defined to determine the degree of user interest is the following:

1. One star: The content is not interesting.
2. Two stars: The content is a bit interesting.
3. Three stars: The content is interesting.
4. Four stars: The content is very interesting.
5. Five stars: The content is essential.

4.2.2 Comments system

This system allows users to comment and say what they think in about the content.

4.2.3 Referrals system

This system allows the user to recommend the content to another user. This could indicate that if the user recommends content to a friend, is because the content is interesting and he believes also it will be interesting for his friend.

4.3 Catching implicit parameters

The fundamental basis of a recommendation system is the ability to collect the data necessary to perform efficiently the feedback process. With the capture of implicit parameters we can measure the user interaction with an electronic book, and we can recover the users' information without their intervention. This process helps to the recommender systems to discover the users interests [6].

When a user is logging, the application collects the data from the iteration of the user, allowing the system to know which categories, content or items the user visited, the time that he spent on each content, how many times he visited each content, comments made by user to the content, content recommended to another users, etc.

One of the most interesting data capture is the capture of reading data, which is that the user is viewing the items (photos) and browsing each, allowing the user to forward or back from one page to another, or back to main page content.

Clicking on the "Next" button the user can advance to the next page (item), the "Back" button the user can return to the previous page as a reader does in a traditional book and the link "Back to album" the user can return to the main content (photo book), this can infer that the user has finished reading the content and closed the book.

Figure 4 depicts an item page (photo), where it shows the photo image and the data related to this item. That is, this part would be the content of the page of a book, in which the user stops reading, underline or write any entry or note, indicating that the user take a while viewing and interacting with each item. These actions by the user interface allows us to measure the parameters defined, and at the same time to store the implicit feedback.

Figure 4: Photo Viewer

4.4 Implicit parameters to measure

As Nielsen [21] presents a series of indicators to measure Web usability and [5] measure and analyze some indicators to predict the implicit interest, we choose and analyze a set of parameters that can help discover the interests of users.

The different parameters measured in the application and whose values are retrieved during user interaction with the application, described below:

- **Duration of the session/content size:** With the evaluation of this parameter indicates the user's connection time, allowing the system know how long it took the user to evaluate and interact with content.
- **Number of clicks:** This parameter will determine how many clicks the user needed to evaluate content.
- **Reading time of a content:** With this parameter will determine how long a user takes reading or viewing a content. This parameter is important because it could determine the user's interest based on the average time reading or viewing content.
- **Number of visits to a content:** this parameter determines the number of times a user read or viewed content. It may determine that a larger number of repetitions, more interest by the content.
- **Reading time of a category:** With this parameter will determine how long a user takes a reading or viewing category o classification.
- **Number of accesses to a category or classification:** This parameter determines how often the user visited a specific category or classification.
- **Number of comments**: With this parameter we could determine the general interest by a specific content, according to the amount of comments that have content.
- **Number of recommendations to a friend:** This parameter determines the interest of users by a content basing on number of recommendations. We can infer that if a user recommended a content is because he has any interest by the content, or he thinks that the content may be of interest to other users.

4.5 Getting data

Once the application distributed to a number of users, we stored on a database the parameters values that will help us carry out our study. In the database we have separate the explicit feedback and the history of actions performed by users (implicit feedback). From these data importantly, the data from the user administrators of the application have been excluded. We define the following interesting relations for our study:

- **Relation I:** Average time of a content visualization versus explicit rating of the content.
- **Relation II:** Number of visits to a content versus explicit rating of the content.
- **Relation III:** Number of comments made to the content versus explicit rating of the content.
- **Relation IV:** Number of recommendations to a friend versus explicit rating of the content.
- **Relation V:** Number of visits to a category or classification versus explicit rating of the contents of that category.
- **Relation VI:** Average rating given to the content according to the number of visited items.
- **Relation VII:** Sequence of visits to a content by each user (sorted in time) versus explicit rating of the content.

4.6 Analysis of data

After having defined and established in the previous section a series of relations between explicit and implicit feedback, in this section we analyze and compare the results. These determine what is the relation between both methods of feedback, determining how effective it may be implicit feedback in recommender systems for electronic books. For each of the established relations we generate the graphs that allow us to better visualize the results.

4.6.1 Relation I: Average time of a content visualization versus explicit rating of the contents.

As shown in Figure 5, we selected a representative sample of the contents that was valued and we compared the explicit rating given by users to each content versus average time spent viewing the content. We can say that the top rated contents are those in which average viewing time is greater. This indicates that a more time viewing at content, the tendency of the user interests is greater.

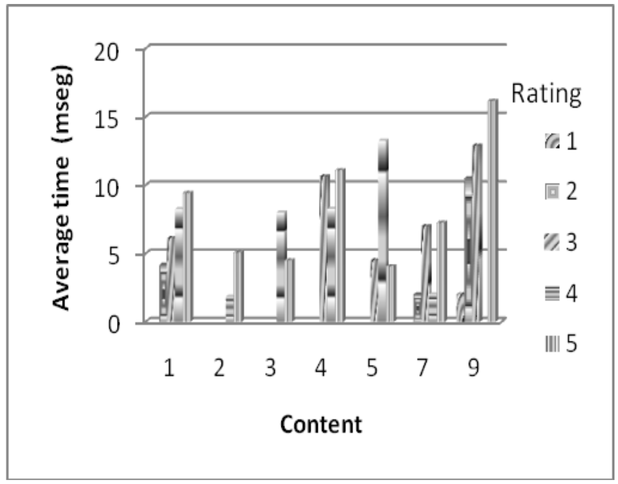

Figure 5: Average time of a content visualization versus explicit rating of the contents

4.6.2 Relation II: Number of visits to a content versus explicit rating of the content.

If now we compare the explicit evaluation versus number of iteration with the contents, as shown in Figure 6, we note that the contents have poorer assessment are those in which the iteration with the same minimal. When viewing these results, we realize that when a user is interested in content tends to redisplay it.

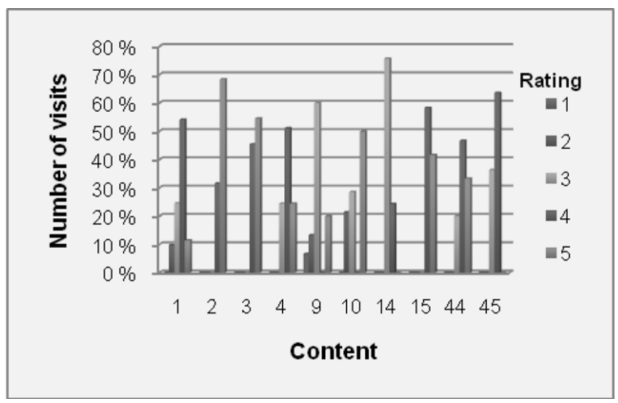

Figure 6: Number of visits to a content versus explicit rating of the content

4.6.3 Relation III: Number of comments made to the content versus explicit rating of the content.

When analyzing the number of comments recorded at the contents and compare with the explicit ratings of the same, we can see in Figure 7 that the ratio of number of comments is directly proportional to the valuation of the contents discussed. Namely, a greater number of comments, higher is the average rating for content.

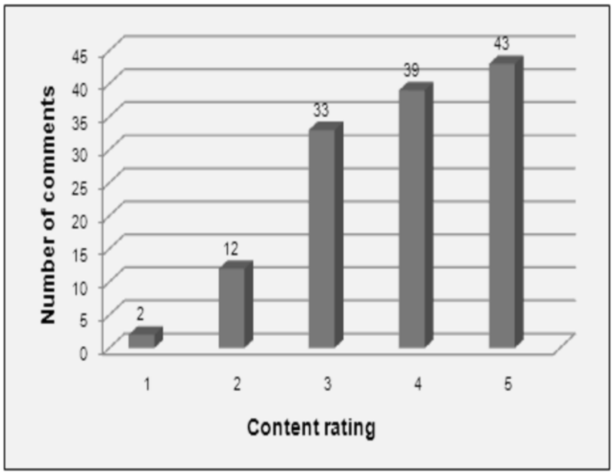

Figure 7: Number of comments made to the content versus explicit rating of the content

4.6.4 Relation IV: Number of recommendations to a friend versus explicit rating of the content.

As shown in Figure 8, we can see that the content that recommend users to other are those in which the user has some interest. These results indicate that when a user is interested in content, he recommends it to their friends. So, we can see that the users only recommend the contents with a positive rate.

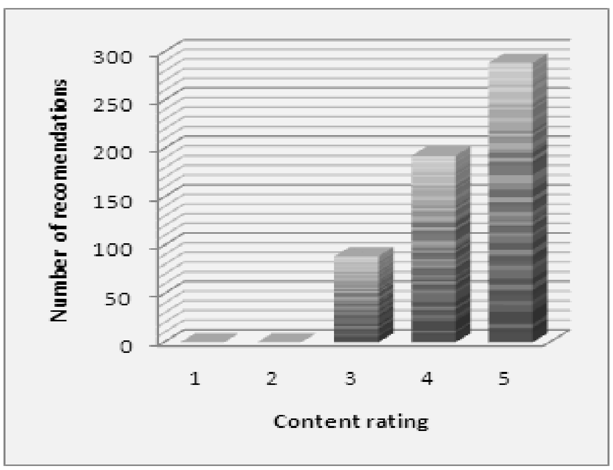

Figure 8: Number of recommendations to a friend versus explicit rating of the content

4.6.5 Relation V: Number of visits to a category or classification versus explicit rating of the contents of that category.

Looking at the results of the visits made to each category and compared with the evaluations made to the contents of this category we can be seen in Table 1 that the content best values are those that belong to the most visited categories. These results indicate that when a user accesses multiple times to a same category is because he is interested in the contents of that category.

Table 1: Number of visits to a category versus explicit rating of the contents of that category

	Rating				
Category	1	2	3	4	5
12	1,00 %	7,96 %	39,80 %	32,34 %	18,91 %
13	0,00 %	10,63 %	15,00 %	25,63 %	48,75 %
14	0,00 %	0,00 %	15,19 %	44,30 %	40,51 %
15	0,00 %	10,47 %	25,58 %	45,35 %	18,60 %
17	0,00 %	14,00 %	26,00 %	19,00 %	41,00 %
18	2,63 %	17,11 %	19,74 %	30,26 %	30,26 %
19	22,73 %	0,00 %	18,18 %	31,82 %	27,27 %
20	15,53 %	15,53 %	17,48 %	22,33 %	29,13 %
22	0,00 %	0,00 %	40,00 %	38,18 %	21,82 %
23	0,00 %	0,00 %	12,05 %	46,99 %	40,96 %
24	0,00 %	21,79 %	24,36 %	21,79 %	32,05 %
25	0,00 %	21,82 %	40,00 %	20,00 %	18,18 %

4.6.6 Relation VI: Average rating given to the content according to the number of visited items.

When we compare the average rating of content versus the number of items that visitors viewed in that content, as shown in Figure 9 we realize that there is not a strong relation between these, due to the dispersion results shown. But despite this dispersion can be noted that the contents in which visitors viewed all item tend to get a better view and so there is a positive tendency.

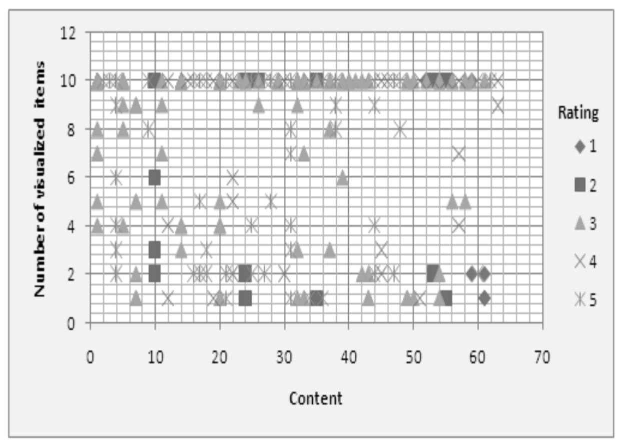

Figure 9: Average rating given to the content according to the number of visited items

4.6.7 Relation VII: Sequence of visits to a content by each user (sorted in time) versus explicit rating of the content.

In this section we will see how users behave on the contents and what changes as they look content randomly or sequentially, pretending to determine whether it is relevant when a user visits the content sequentially or when it does randomly.

In general, it has been observed that when users watch the content at random, the valuations fluctuate more than when user watching it sequentially.

To illustrate this, we show data for three different users chosen as representative cases. The user 13 shows a sequential trend when visiting content, while user 4 shows a random trend, and finally, the user 25 to check sequential intervals interspersed with intervals of random visits.

For each of them shows a figure that represents the sequence of visited content over time and other ratings for this content, also situated chronologically, so that comparison easier.

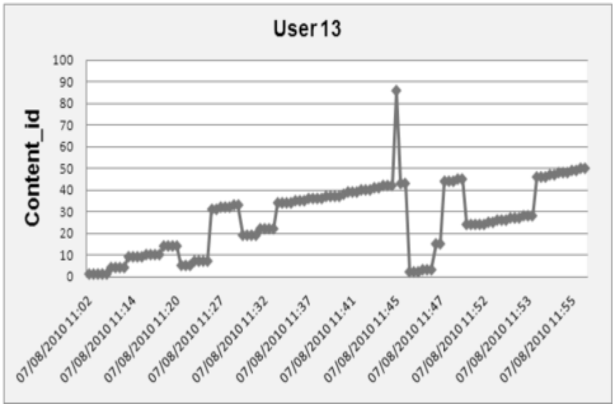

Figure 10: Sequence of the contents visited by the user 13 (sorted in time)

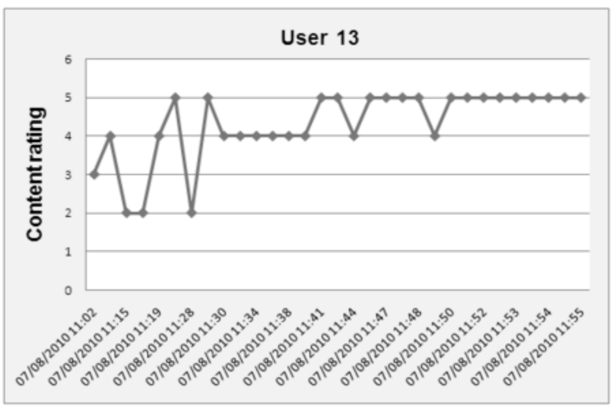

Figure 11: Sequence of the contents rating by the user 13 (sorted in time)

Figure 10 shows as the user 13 tend to visit the content sequentially, as the contents IDs are incremented by one in most of the time. Compared with Figure 11, we observed that the ratings remain fairly constant in the sequences of content.

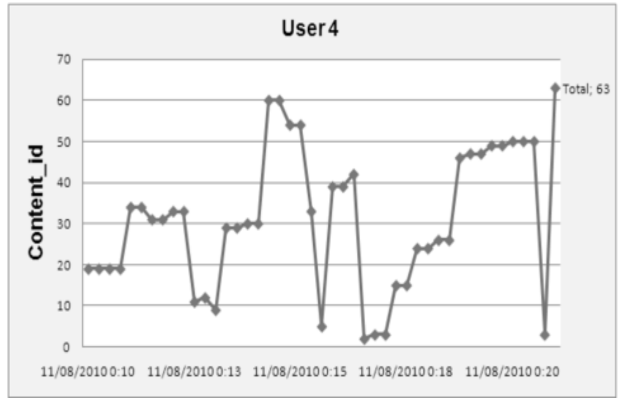

Figure 12: Sequence of the contents visited by the user 4 (sorted in time)

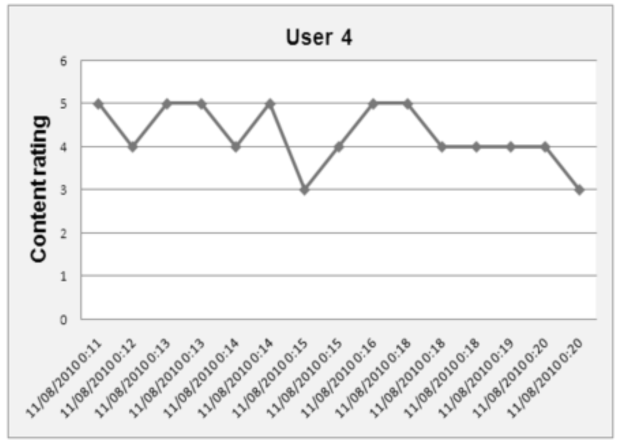

Figure 13: Sequence of the contents visited by the user 4 (sorted in time)

Figure 12 shows the visits of user 4 to the different content at over time. It is clearly seen that user does not follow any apparent order in navigation, namely, He did not visit consecutive contents. When we compared with the assessments given by the user to the contents, which are shown in Figure 13, we observe that vary much more than in the case of sequential visits.

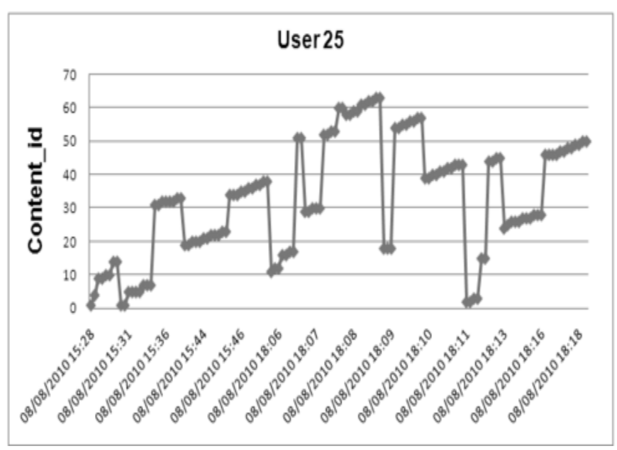

Figure 14: Sequence of the contents visited by the user 25 (sorted in time)

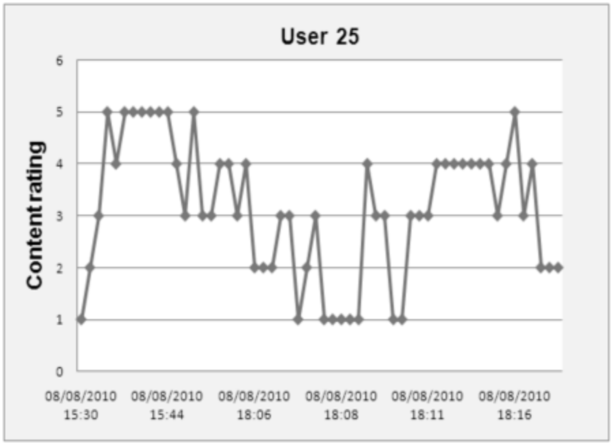

Figure 15: Sequence of the contents rating by the user 25 (sorted in time)

Finally, we note the user 25, because this is a very active user in the system that blends perfectly the random visits with sequential visits.

In Figure 14 we observe as user visits several contents correlated and then he jumps to another series of correlated contents. These correspond to the categories navigation: the user browses the contents of a category and then he goes to see the contents of another different category. In Figure 15, we see the line of ratings over time has many jumps, but notes that, in general, the jumps occur when sequences of user visits to the content is interrupted.

4.7 Final results

After observing and performing the analysis of the relations established in this section we stand out the final results obtained during the process.

In principle, one of the main aspects discussed is the time spend by the users on each option in the application. So, we see that the average time display of the content is related to the valuation given to that content, and the interest is higher in the contents best rated.

When we relate the number of times a user visits content with the explicit ratings of this content, we observed that contents that a user saw in more than one occasion and has even saw their items several times; these contents have a high valuation. It follows that content that the user repeat is because he likes it.

When studying the previous relation is also observed that most ratings given by users to contents is high, indicating that either the users access to content that they dislike, or they do not rate the contents that they dislike. The first situation may be due to classify content into categories: this way the user can make a first selection between what contents he likes and what contents he does not likes. In the same direction, we can see that when a user accesses several times to the same category is because he is interested in the contents of that category. The second case is explained by the optimism of the user: users are much more likely to "publish" they like something that to say that they do not like something.

On the other hand, when we relate the number of comments with the content rating, we determined that the number of comments is directly proportional to the rating, indicating that users tend to comment the contents that they are interested, or when the number of comments in a content is greater, the users tendency to comment this content is greater.

Moreover, if we observe the relation between recommended contents and explicit ratings, we observe that the recommended contents have a positive rating, so we can say that users only recommends the interesting contents.

When we look at the items count of each content visualized by the users, it appears that there is not exact pattern that defines the relation between the item number and explicit ratings. But despite this dispersion can be noted that the contents visualized completely by users tend to get a better rating. This is because when a user likes a item in a content normally displays all its items, or by the user intuition that he believes that, if he likes several items of a content also he likes the underlying items.

Finally, it is interesting to see if the order in which the user sees the content and the relation with the explicit ratings. In the data presented above, we observed that when users visit a sequential content, such content ratings are also uniform. This indicates that if a user likes a content then the contents near of this also likes.

5 CONCLUSIONS AND FUTURE WORK

After measuring the value of the implicit parameters defined in this study, analyzing and comparing the grade of correlation between explicit and implicit feedback, we have reached a series of conclusions through which more effective recommender systems can be built, mostly based on the user's behavior. Summing up, we can assert that:

- The more time a content is displayed by a user, the more he likes it and therefore the higher he rates it. So there is a direct relation between displaying time and explicit ratings.
- The more a user visits a content or category, the more he is interested on it. So there is a direct relation between the number of visits and explicit ratings.
- When a user accesses multiple times a category it is because he likes the contents of that category, so the categorization of contents have an influence the user's interests.

- There is not a strong relation between the display all of the items of a content and its explicit ratings, but there is a positive trend.
- There is certain inertia in the comments: already commented contends tend to acquire more comments.
- When a user comments content it is because he has any kind of interest on it.
- Users explicitly recommend the contents that he finds interesting.
- Content with a high average rating is not meant to have been visited many times.
- If a user is satisfied by content while visiting it, adjacent contents will probably satisfy them too.

In addition to the parameters studied and analyzed in this paper, the next stage of this project will integrate other user parameters that can also be measured, based in actions such as: content sharing on social networks, content printing, sending contents by e-mail. We will also capture other special parameters such as highlights, annotations made in the document, etc. These parameters will be obtained with the application for e-book reader that is being developed in **eInkPlusPlus**. And finally, make algorithms to transform the information about implicit feedback into explicit ratings that existing recommender systems based on explicit ratings can use.

ACKNOWLEDGMENT

This work was performed by the University of Oviedo under Contract No. TSI-020110-2009-137 of the research project **eInkPlusPlus**.

6 REFERENCES

[1] O. Sanjuan Martínez, Pelayo G-Bustelo,C., González Crespo, R., Torres Franco, E., "Using Recommendation System for E-learning Environments at degreelevel," *International Journal of Artificial Intelligence and Interactive Multimedia*, vol. 1, pp. 67-70, 2009.

[2] J. O'Donovan and B. Smyth, "Trust in recommender systems," presented at the Proceedings of the 10th international conference on Intelligent user interfaces, San Diego, California, USA, 2005.

[3] N. Taghipour and A. Kardan, "A hybrid web recommender system based on Q-learning," presented at the Proceedings of the 2008 ACM symposium on Applied computing, Fortaleza, Ceara, Brazil, 2008.

[4] S. Noor and K. Martinez, "Using social data as context for making recommendations: an ontology based approach," presented at the Proceedings of the 1st Workshop on Context, Information and Ontologies, Heraklion, Greece, 2009.

[5] M. Claypool*, et al.*, "Inferring User Interest," *IEEE Internet Computing*, vol. 5, pp. 32-39, 2001.

[6] E. R. Núñez Valdéz.*, et al.*, "First Steps towards Implicit Feedback for Recommender Systems in Electronic Books," in *Distributed Computing and Artificial Intelligence*. vol. 79, A. de Leon F. de Carvalho*, et al.*, Eds., ed: Springer Berlin / Heidelberg, 2010, pp. 61-64.

[7] P. Resnick*, et al.*, "GroupLens: an open architecture for collaborative filtering of netnews," presented at the Proceedings of the 1994 ACM conference on Computer supported cooperative work, Chapel Hill, North Carolina, United States, 1994.

[8] R. González Crespo*, et al.*, "Recommendation System based on user interaction data applied to intelligent electronic books," *Comput. Hum. Behav.*, vol. 27, pp. 1445-1449, 2011.

[9] P. Wang, "Why recommendation is special?," *Workshop on Recommender Systems, part of the 15th National Conference on Artificial Intelligence*, pp. 111-113, 1998.

[10] P. Resnick and H. R. Varian, "Recommender systems," *Commun. ACM*, vol. 40, pp. 56-58, 1997.

[11] G. Adomavicius*, et al.*, "Incorporating contextual information in recommender systems using a multidimensional approach," *ACM Trans. Inf. Syst.*, vol. 23, pp. 103-145, 2005.

[12] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Trans. on Knowl. and Data Eng.*, vol. 17, pp. 734-749, 2005.

[13] L. Terveen*, et al.*, "PHOAKS: a system for sharing recommendations," *Commun. ACM*, vol. 40, pp. 59-62, 1997.

[14] H. Kautz*, et al.*, "Referral Web: combining social networks and collaborative filtering," *Commun. ACM*, vol. 40, pp. 63-65, 1997.

[15] M. Balabanovi*, et al.*, "Fab: content-based, collaborative recommendation," *Commun. ACM*, vol. 40, pp. 66-72, 1997.

[16] G. Linden*, et al.*, "Amazon.com recommendations: item-to-item collaborative filtering," *Internet Computing, IEEE*, vol. 7, pp. 76-80, 2003.

[17] C.-N. Ziegler*, et al.*, "Improving recommendation lists through topic diversification," presented at the Proceedings of the 14th international conference on World Wide Web, Chiba, Japan, 2005.

[18] G. Jawaheer*, et al.*, "Comparison of implicit and explicit feedback from an online music recommendation service," presented at the Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems, Barcelona, Spain, 2010.

[19] D. Kelly and J. Teevan, "Implicit feedback for inferring user preference: a bibliography," *SIGIR Forum*, vol. 37, pp. 18-28, 2003.

[20] Y. Hu*, et al.*, "Collaborative Filtering for Implicit Feedback Datasets," presented at the Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, 2008.

[21] J. Nielsen, "Usability Metrics: Tracking Interface Improvements," *IEEE Softw.*, vol. 13, pp. 12-13, 1996.