Computers in Human Behavior xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

Creating recommendations on electronic books: A collaborative learning implicit approach

Edward Rolando Núñez-Valdez [a,*], Juan Manuel Cueva Lovelle [b,1], Guillermo Infante Hernández [b,1], Aquilino Juan Fuente [b,1], José Emilio Labra-Gayo [b,1]

[a] University Carlos III of Madrid, Computer Science and Engineering Department, Avenida de la Universidad, 30, 28911 Leganés, Madrid, Spain
[b] University of Oviedo, Computer Science Department, Sciences Building, C/Calvo Sotelo s/n, 33007 Oviedo, Asturias, Spain

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Recommender system
Books recommendation
Implicit feedback
Collaborative learning

ABSTRACT

Recommender systems appear among other reasons with the purpose to improve web information overload and ease information recovery. This kind of systems aid users to find contents in a non-difficult way and with minimal effort. Even though, a great number of these systems performance requires contents to be explicitly rated in order to determine user's interest. When interacting with electronic books this performance may alter users reading and understanding patterns as they are asked to stop reading and rate the content. Therefore, the analysis of user behavior, preferences and reading background can be considered suitable for a recommender system to build collective web knowledge in a collaborative learning context. This way, recommender system can assist users in finding contents of their interest without explicit rating based on previous constructed knowledge. The goal of this research is to propose an architecture to build a content recommendation platform based on eBook reading user behavior, allowing users to learn about the digital content collaboratively. This platform is formed by web readers' community that aids members in finding contents of their interest in an automatic way and with minimal effort.

1. Introduction

Last years, the exponential growing of the information available on the web brought an issue called information overload. Hence, the great data amounts makes difficult to discover, find and classify the most relevant information for each user profile or interests (Zhang, Zhou, & Zhang, 2011). Commonly, users seek for recommendations from another users or media in order to find the most valuable information or products they need (González Crespo et al., 2010; Su & Khoshgoftaar, 2009). Recommender systems are usually employed to deal with information overload on the web as an information recovering and classification technique. They filter the information available on the web and help users to find more interesting and valuable information (Noor & Martinez, 2009; O'Donovan & Smyth, 2005; Taghipour & Kardan, 2008). The most relevant search engines like Google, or online stores like Amazon, have incorporated recommender technologies as part of their services with the purpose to personalize the search results (Verbert et al., 2012).

Despite the major upswing and extensive utilization of these systems, there is a gap in the information feedback process, which is a key part of all the recommendation process that is susceptible of improvement. This paper sustains that recovery, analysis and transformation of user behavior can be used to measure their interest in some determined contents and therefore be able to bring more accurate recommendations to them. Even though, as illustrated in (Claypool, Brown, Le, & Waseda, 2001), the most common solutions and the more prevalent are the ones based on explicit ratings. In the context of eBooks these techniques can alter the user's regular navigation and reading patterns, because they have to stop and rate the items.

In (Nuñez-Valdéz et al., 2012), it was recently defined a set of implicit parameters on which was performed a comparative analysis that led to the correlations between the actions that a user can perform during an eBook reading time and the explicit ratings given by it on each content. These findings showed that is possible to determine user interest through the analysis and transformation of its behavior. Taking into account these results and, with the implementation of an architecture that contains an algorithm to perform this transformation, recommender systems can be constructed in a more precise manner, based on implicit feedback.

* Corresponding author. Tel.: +34 916249109.
E-mail addresses: ednunezv@inf.uc3m.es (E.R. Núñez-Valdez), cueva@uniovi.es (J.M.C. Lovelle), infanteguillermo@uniovi.es (G.I. Hernández), aajuan@uniovi.es (A.J. Fuente), labra@uniovi.es (J.E. Labra-Gayo).
[1] Tel.: +34 985182451, +34 985103397; fax: +34 985181986.

http://dx.doi.org/10.1016/j.chb.2014.10.057
0747-5632/

In these times of information overload, emerges the necessity to develop recommender systems which allows discovering users' interest in a more effective and simple way improving their experience and satisfaction. The possibility of analyzing and studying the users behavior on a social network of electronic books allows us to improve the collaborative learning of its members. The use of recommendation systems allows readers to create and share collective knowledge in an easily and automatically way.

The rest of this paper is structured as follows Section 2 presents the background of recommender systems; Section 3 shows a case study and the architecture proposed; Section 4 presents the evaluation of results; and finally, in Sections 5 and 6 are explained the future research directions and conclusions of this work, respectively.

2. Background

Recommender systems are tools that aid users to find the information they really need in an easy and efficient manner. These systems helps to optimize the time users employ in searching contents that somehow are harder to find. These contents are selected by recommender systems from a large amount of data that is available on the web and can be any kind, such as books, movies, songs, websites, blogs (González Crespo et al., 2010).

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) (Resnick & Varian, 1997). These systems not only aid users in finding contents of their interest, but also contribute in a certain way to the development of enterprises that use them. Once the users can access the contents in an easy way, they are more likely to buy products or services which increase sales and help entrepreneurs to improve their marketing strategies.

As shown in (González Crespo et al., 2010; O'Donovan and Smyth, 2005; Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994), recommender systems try to solve the issue of data overload, facilitating access to the vast amount of information available on the web through the implementation of algorithms and classification mechanisms of information. Nevertheless, when a recommender system does not have enough information about a user or content, becomes difficult to perform recommendations and particularly valid ones.

This problem arise because some recommender systems type (e.g. collaborative filtering) present some problems like: (1) The cold start problem which come from the contents that nobody has rated yet both explicitly or implicitly, over a data set (Schein, Popescul, Ungar, & Pennock, 2002); (2) Sparsity problems that occurs when available data are insufficient for identifying similar users (Papagelis, Plexousakis, & Kutsuras, 2005); (3) New item problem that takes place when an item that has not been previously rated by any user, it is not considered by the system (Adomavicius & Tuzhilin, 2005); and (4) Popularity bias problem which states that different items cannot be recommend to someone with a unique taste.

Recommender systems can be classified into different types according to the type of information used to make recommendations (Adomavicius, Sankaranarayanan, Sen, & Tuzhilin, 2005). Traditionally there are several paradigms of filtering information used to generate recommendations and these are classified as: (1) Collaborative filtering: calculates the similitude between users and creates the called "close neighbors" for making recommendation; (2) Content-based: try to recommend similar content to a particular user, based on content that to another user liked in the past; and (3) Hybrid approach: is the combination between content-based and collaborative filtering.

Other variety of techniques have been proposed for performing recommendation by other authors as (Adomavicius & Tuzhilin, 2005), 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 wide ranges of recommendation systems that are used in different areas, whether for commercial or scientific or experimental purposes. For example: PHOAKS (Terveen, Hill, Amento, McDonald, & Creter, 1997), Referral Web (Kautz, Selman, & Shah, 1997), Fab: content-based collaborative recommendation (Balabanović & Shoham, 1997), Amazon.com recommendations: item-to-item collaborative filtering (Linden, Smith, & York, 2003).

Recently, other proposals have been presented, such as (Montes-García, Álvarez-Rodríguez, Labra-Gayo, & Martínez-Merino, 2013) where the authors present a hybrid news recommendation system that introduces a context-aware feature for journalists to enable the identification of similar topics across different sources. In (Lee & Park, 2007) a mobile web news recommendation system is presented for making recommendations using the mobile content and the web news services.

Through feedback information techniques, a recommender system should be able to gather the most quantity of information related to a user's profile as possible. This allows to discover users' preferences and interests by determined contents to later generate more accurate recommendations as shown in (Adomavicius et al., 2005; Pommeranz, Broekens, Wiggers, Brinkman, & Jonker, 2012; Resnick & Varian, 1997; Ziegler, McNee, Konstan, & Lausen, 2005). These techniques are classified into two types: (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. For example, among the most common explicit recommender systems used on the web can be found the following: *star ratings system* used by Amazon online store and film affinity; *Like rating system* used by social networks as Facebook and YouTube. (2) Implicit feedback: This process consists on evaluating 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. These techniques take advantage of user behavior to understand user interests and preferences (Kelly & Teevan, 2003). The use of this feedback technique helps to improve the user's experience and satisfaction when searching contents over the Web since it does not requires explicit ratings to receive recommendations.

Nowadays, the majority of the study cases and implemented recommender systems normally use feedback mechanisms based on explicit feedback, however this may be inconvenient to users, as they typically do not like to rate contents. As stated in (Claypool et al., 2001), explicit ratings are the most common and obvious indicators of the user's interest, because it allows them to tell the system what they really think of the rateable objects. On the another hand, they alter the user's regular navigation and reading patterns, because they have to stop and rate the items. In this sense, implicit feedback techniques seems to be an attractive candidate to improve the information recovery mechanism as there is not required a further effort from the user (Kelly & Belkin, 2001).

3. Case of study

One of the main issues of recommender systems is the deficit on the implementation of information feedback mechanisms. The main reason of this deficit in most of the cases takes place because of these mechanisms are based on explicit feedback which can be an inconvenient for users as they usually do not like to rate contents. Hence, if users do not rate contents it is not possible to know the contents of their interest, reason why it cannot be possible either to recommend contents to them by their profile using a recommender system. From this mentioned issue, it becomes necessary to gather the most quantity of information related to the behavior and reading habits from users' interaction in an implicit way. This way their interests and needs can be determined which makes possible to implement a more efficient feedback mechanism. This mechanism can improve recommender systems functionalities.

In order to achieve an approximation to the solution of explicit feedback in recommender systems, an architecture that allows to analyze and transform users' behavior in approximate explicit ratings it is proposed and implemented. The implementation context of this architecture is the eBooks environment. These approximate explicit ratings can be used by any recommender engine that functions with this sort of data. To validate the proposed architecture, each one of the modules or systems that forms it were developed and a mathematical transformation model was defined in order to analyze the main actions that define users' behavior in an eBooks platform. This model is later implemented and evaluated though a number of tests.

3.1. Architecture proposed

Fig. 1 depicts the high-level architecture design for an eBooks recommender platform based on users' behavior. This architecture is formed by three main levels: client applications level that allows users to interact with the platform; intermediate level composed by a feedback system which can obtain the information from users and, an explicitation system that performs the analysis and conversion of implicit information into explicit values; the third level persists the implicit information obtained during feedback, the resulting explicit information from the explicitation process and, the configuration files that contains the meta-information from analyzed actions and other parameters. With data processed and analyzed, a recommender engine that recommends, despite the redundancy, contents to users based on their profiles, is implemented.

The proposed architecture enables the implementation of a recommender system for eBooks based in implicit feedback. This architecture allows building a collective web of knowledge to facilitate collaborative learning on the web.

3.2. Architecture implementation

There is some common actions users perform when they use eBooks, e.g.: read, share, recommend, annotate, remark, browse by contents, etc. Analyzing these relevant basic actions or (basic behavior) for content recommendations within a social network can be found the actions that plays a key role in discovering users interest. Table 1 shows a set of commonly performed actions from eBook platform users. The study of these actions allows evaluating users' behavior and determining their interests in some specific contents. As shown in Fig. 2. It has been designed and developed the EBook Contents Recommender Platform (ECRP) with the purpose to get an approximation to the explicit feedback solution in eBook recommender systems. This platform enables to recommend digital contents based on users' interest by analyzing their behavior and reading habits.

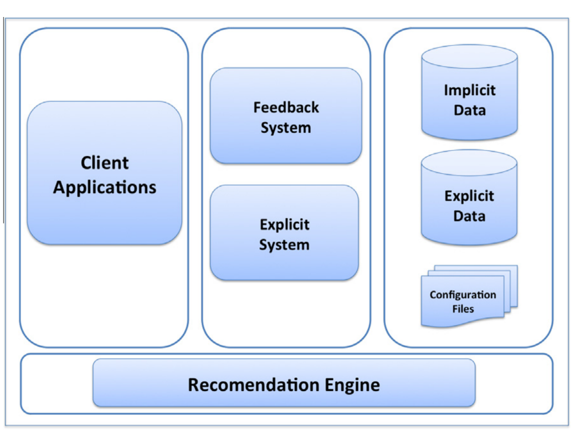

Fig. 1. High level architecture.

This architecture has been designed with the previous high level defined structure in mind (see Fig. 1). This time it is given a more detailed view of the components. Fig. 2 describes ECRP which implements all the components defined in the high level architecture. It can be observed that ECRP captures users' behavior related information through a feedback system. This data can be obtained from a web client or a mobile device communicated with the platform by web services. Later, data is used by the explicitation system that analyzes users' behavior and determines the interest level in contents they have interacted with, (transform implicit in explicit data). Finally with the recommender engine implementation, explicit data is registered and recommendations to interest users are made.

The main goal pursued with the implementation of this platform is to demonstrate the technical feasibility and practical utility of the proposed architecture. With that purpose, an implementation that includes the integrant applications of the proposed platform have been created and depicted as follows:

3.2.1. Web client

The objective of this architecture module is to define a web application that allows users to discover and share contents within an eBook readers community. This web application takes the form of a social network and is called *eInkPlusPlus*. It is a social network that enables contents sharing and management between registered users. It eases contents diffusion and access using web browsers and mobile devices. The development of this social network seeks for an evolution in users' interacting activities when using eBooks, offering smart digital contents adapted to the needs of each user.

3.2.2. eBook client

The goal of this module is to define an eBook reader that enables users to interact with shared contents in an efficient manner. This module eases the contents reading and the performing of other actions over the content. These actions can include: contents remarks, annotations, sharing, etc., that belongs to the developed prototype called *eInkPlusPlus Reader. It* is an eBook reader for mobile devices developed for Android systems that enables users to read contents from the platform, as well as synchronize its contents available in the eInkPlusPlus social network.

3.2.3. Feedback system

This module represents an application development that enables to register the actions performed by user in an efficient way. The goal to reach here is to do a later analysis of users' behavior that can be easily configured in social platform. To gather the most possible quantity of information during users' interactions, a Ruby on Rails *User Interactions Recorder* (*UIREC*) was developed. This recorder allows recollecting and storing users' actions in an implicit way by using the web platform or a smart mobile device through web services. With this implementation, when a user performs an action (request) from the web or from eBook reader,

Table 1
Most common actions that define the behavior of the users in an electronic book platform.

Id	Name	Type	Indicator	Weight	Scope
A_1	Explicit rating of a content	Explicit	–	–	Individual
A_2	Time spent reading a content	Implicit	Positive	0.1	Social
A_3	Highlighting a content	Implicit	Positive	0.1	Social
A_4	Adding a note to a content	Implicit	Positive	0.1	Social
A_5	Commenting a content	Implicit	Positive	0.1	Social
A_6	Recommending a content to a contact	Implicit	Positive	0.1	Individual
A_7	Adding a content to the collection	Implicit	Positive	0.1	Individual
A_8	Adding a content to the list of favorites	Implicit	Positive	0.1	Individual
A_9	Rejecting a content recommendation	Implicit	Negative	0.1	Individual
A_{10}	Eliminating a content from the list of favorites	Implicit	Random	0.1	Individual
A_{11}	Eliminating a content from the collection	Implicit	Random	0.1	Individual

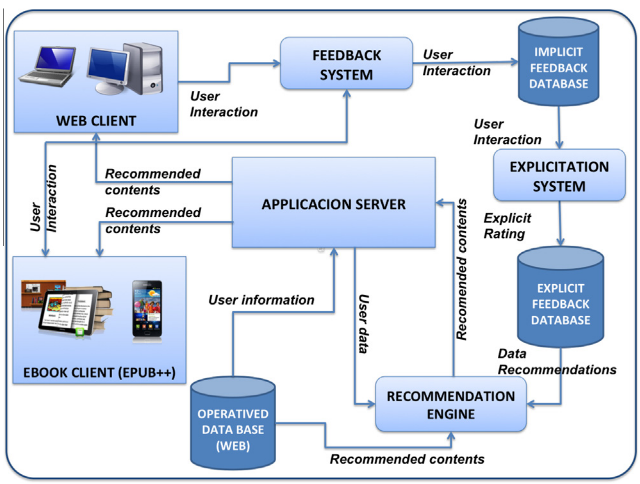

Fig. 2. Platform architecture implementation.

e.g. to do an annotation, remark, comment, etc., a controller is invocated and it executes the necessary logic.

3.2.4. Database systems

Database systems are formed by three important databases:

- Implicit Feedback Database stores all the information resulting from users' interaction with the social network and eBook reader.
- Explicit Feedback Database stores data resulting from data explicitation processes.
- Operative Database (Web) stores data resulting from data operative data on the web platform, as well as the recommender engine's generated recommendations.

3.2.5. Explicitation system and recommendation engine

Explicitation system deals with the transformation of the implicit feedback to explicit ratings. In order to evaluate the different users' behavior according to their reading habits and interaction with the platform, a *User Interactions Converter Algorithm* (*UICA*) is developed. This algorithm evaluates implicit actions previously configured on the platform using a set of procedures to convert these values in explicit ratings. These ratings are generated within previously established range that indicates users' interest. Table 1 shows a set of user's commonly performed actions within an eBooks platform that has been evaluated with the implementation of this platform.

The implementation of explicitation system consists on an application that obtains the implicit data (interactions between users) and converts it into explicit ratings through the implementation of a converter algorithm (UICA). As seen in (Nuñez Valdéz, Cueva Lovelle, Sanjuán Martínez, Montenegro Marín, & Infante Hernandez, 2011) users usually rate a content by using explicit rating systems, such as the "five stars" system or the "like/don't like" system, with which they reflect their interest for that content. The *User Interactions Converter Algorithm* (*UICA*) evaluates the selected actions and converts them into an explicit value, that is to say that it performs an action depending on the explicit rating that the user gives to content.

In the following section we define the mathematical conversion model that specifies the different formulas that allow to calculate the value of the actions performed by the users and which also served as a base for the development of the UICA. In order to define the mathematical conversion model, the elements from Table 1 it is explained, with the purpose of identifying and clarifying the characteristics of the actions in a clearer and more precise way: **Id** represents the action's identifier; **Name** is the name of the action that is being performed by the user in the platform; **Type** shows the kind of feedback mechanism the action (implicit or explicit) belongs to; **Indicator** is a classification that adds a default value to the interest of the users when they perform this action; **Weight** is the level of importance of each action in relation the other actions of the same type. It must be pointed out that the sum of the weights of all those actions should always be 1. On the other hand, the explicit action that is shown in Table 1 does not have weight because the user itself defines its value, and so no conversion process is performed on it and **Scope** shows that the action's value is calculated by taking into consideration the behavior of the other users of the platform.

Finally, recommendation engine consist of a recommender engine implementation based on generated data from the explicitation system, which recommends contents that can be interesting for users.

3.2.5.1. Mathematical conversion model. For UICA to obtain a rating that represents the interest of an *i*-th user for a *j*-th content based on the analysis and interpretation of the actions that the user performs around content, several mathematical equations were defined. The aim for implementing these equations is to study the behavior of the user for each performed action and converting it into a numerical value defined within a certain range.

This default range is defined with the objective of simulating the explicit rating of a content, that is to say, if the "five stars" system was used the range could be (1…5), and if the "Like/Don't Like" system was used the range could be (1…2). This means that the inferior value would be the worst rating the user would give to the content, while the superior value would be the best rating. A zero (0) value means that the user has not rated the content yet.

3.2.5.2. Calculation of the final rating of a content. The final rating of a *j*-th content for an *i*-th user is determined by measuring each action separately and giving a *P* weight to it. The *P* weight assigns a level of importance to each action when calculating the user's interest. The calculation shows that if the content is rated explicitly (A_1), the rating will be equal to the value given by the user. Otherwise, the rating will be equal to the implied actions calculation ($A_2 \ldots A_k$). The final rating for the interest of the i-th user on the j-th content, based on the user's behavior, is calculated through this equation:

$$V(i,j) = \begin{cases} A_1, & \text{if } A_1 > 0 \\ S, & \text{if } A_1 = 0 \end{cases}$$

where

$V(i,j)$ is the rating of the j-th content for the i-th user.

i the i-th user that performed an action around the j-th content.

j the j-th content around which the i-th user performed an action.

A_1 the explicit rating of the j-th content assigned by the i-th user.

S the value obtained by calculating the implicit actions, which is obtained through the following equation:

$$S = \frac{\sum_{k=2}^{n}(P_k + \mathrm{Pr})A_k + A_k}{N+1}$$

where

P_k is the weight assigned to the A_k action. P_k has to meet the following restrictions:

$$0 \leqslant P_k \leqslant 1$$

$$\sum_{k=2}^{n} Pk = 1$$

k is the sub-index that identifies the action. This variable starts in 2 because this calculation only considers implicit actions.

$(P_k + P_r)\, A_k$ is the percentage of weight added to the value of the action.

N is the amount of actions with the j-th content performed by the i-th user. This value is obtained through the equation:

$$N = \sum_{k=2}^{n} f(A_k)$$

where

$f(A_k)$ is the function that shows that the i-th user performed the A_k action in the j-th content. The value of this function is determined through:

$$f(A_k) = \begin{cases} 1, & \text{if } A_k > 0 \\ 0, & \text{if } A_k = 0 \end{cases}$$

Pr is the remaining weight of the $A_2 \ldots A_n$ actions NOT performed by the i-th user around j-th content which is redistributed between the P_k weights of the performed actions. The Pr value is calculated through:

$$\mathrm{Pr} = \frac{\sum_{k=2}^{n} Q(A_k)}{N}$$

where

N is the amount of actions performed by the i-th user around the j-th content. This value is obtained through the formula defined in the previous paragraphs.

$Q(A_k)$ is the function that returns the value of the A_k action's weight that the i-th user did not performed around the j-th content. The value of this function is determined through:

$$Q(A_k) = \begin{cases} \mathrm{P}, & \text{if } A_k = 0 \\ 0, & \text{if } A_k > 0 \end{cases}$$

3.2.5.2.1. Calculation of the rating of user behavior actions for an electronic book platform. Detailed below are the actions shown in Table 1 and also the mathematical formalization that allows to determine the value for each user action in the feedback process and convert these actions into an explicit rating.

- **A_1 – Explicit rating of a content**

When a user explicitly rates content, the other actions he performed on it are discarded, because the user is showing his interest on that content in an explicit way. This indicates that one of the main points is to know if the user has explicitly rated the content, so when measuring the user's implicit interactions it must be known if that content has been rated previously and if that rating was explicit or implicit. If the content has a previous rating automatically calculated by the system (that is to say, if this rating has been obtained through the analysis and calculation of the actions based on the behavior of the user) and the user rates the content again but in an explicit way, then this new value replaces the previous one, as the explicit rating directly shows the interest of the user for the content. Given two different ratings from a user for the same content (one explicit and other implicit), the final result of the content's rating is equivalent to the explicit rating, regardless of the order and the moment in which those ratings are obtained. The explicit rating of content is obtained through the following equation:

$$A_1(i,j) = x$$

where

i is the i-th user of the platform that explicitly rated the content.

j the j-th content of the platform that was explicitly rated by a user.

x is the explicit rating that the i-th user gave tho the j-th content.

- **A_2 – Reading time of a content**

As it can be seen in (Nuñez-Valdéz et al., 2012), the longer the time spent reading a content the higher the chances that the user is interested on it. Thus, to establish a proper relationship between the time spent on the reading of the content and the real time spent reading the whole content, it is necessary to compare this time with the time that the other users of the platform spent on reading the same content.

To determine the reading's value we must know how much time did the user spent reading each chapter of the book. Measuring the reading by chapters is a better option tan measuring by pages since the amount of these can change depending on the device that is being used. That is because the electronic books automatically adapt their contents to the screen size of the device. The value for the time spent on reading content is obtained through the following equation:

$$A_2(i,j) = \frac{\sum_{k=1}^{n} T_k(i,j)}{n}$$

where

$A_2(i,j)$ is the time spent on reading the j-th content by the i-th user.

i the i-th platform user that read a content.

j the j-th content of the platform read by a user.

n the total amount of chapters that the j-th content has.

$T_k(i,j)$ the normalized value for the time spent on the reading of the k-th chapter of the j-th content that was read by the i-th user. This value is calculated through the following equation:

$$T_k(i,j) = \begin{cases} \frac{\mathrm{Tt}_k(i,j)}{\mathrm{SocialValue}(\mathrm{TTt}_k(i,j))} * (\mathrm{Ls} - \mathrm{Li}) + \mathrm{Li}, & \text{if } \mathrm{Tt}_k(i,j) > 0 \\ 0, & \text{if } \mathrm{Tt}_k(i,j) \leqslant 0 \end{cases}$$

where

Ls is the superior limit of the value normalization of $T_k(i,j)$. See Section 3.2.5.3.

Li the inferior limit of the value normalization of $T_k(i,j)$. See Section 3.2.5.3.

$\mathrm{Tt}_k(i,j)$ the total amount of time spent by the i-th user on the k chapter of the j-th content. This total is calculated through the equation:

$$\mathrm{Tt}_k(i,j) = \sum_{x=1}^{n} x(i,j)$$

where

x is the time spent on the reading of the k chapter by the i-th user on the j-th content.

n represent the different times or reading sessions duration spent on the reading of the k chapter by the i-th user on the j-th content.

$\mathrm{TTt}_k(i,j)$ the total amount of time that each i-th user have spent on reading the k chapter from the j-th content. This is defined as:

$$\mathrm{TTt}_k(i,j) = \{\mathrm{Tt}_k(1,1), \mathrm{Tt}_k(2,1), \mathrm{Tt}_k(3,1), \ldots, \mathrm{Tt}_k(n,1)\}$$

$\mathrm{SocialValue}(\mathrm{TTt}_k(i,j))$ is the maximum, average or median reading time that the i-th user have spent on reading the k chapter from the j-th content, within $TTt_k(i,j)$. The social value's calculation option is selected depending on the configuration:

$$\mathrm{SocialValue}(\mathrm{TTt}_k(i,j)) = \begin{cases} \mathrm{MAX}(\mathrm{TTt}_k(i,j)) \\ \mathrm{AVERAGE}(\mathrm{TTt}_k(i,j)) \\ \mathrm{MEDIAN}(\mathrm{TTt}_k(i,j)) \end{cases}$$

- **A_3 – Highlighting a content**

When reading content, users usually highlight fragments of the text with different colors, giving them different levels of importance. This is action is commonly performed by the user when he wants to highlight words, phrases or even paragraphs from the content that he finds interesting. The value for the highlighting of content is calculated through the following equation:

$$A_3(i,j) = \begin{cases} \frac{\mathrm{Th}(i,j)}{\mathrm{SocialValue}(\mathrm{TTh}(i,j))} * (\mathrm{Ls} - \mathrm{Li}) + \mathrm{Li}, & \text{if } \mathrm{Th}(i,j) > 0 \\ 0, & \text{if } \mathrm{Th}(i,j) \leqslant 0 \end{cases}$$

where

$A_3(i,j)$ is the value for the highlighting of the j-th content by the i-th user.
i the i-th user of the platform that highlighted a content.
j the j-th content of the platform highlighted by a user.
Ls the superior limit of the value normalization of $A_3(i,j)$. See Section 3.2.5.3.
Li the inferior limit of the value normalization of $A_3(i,j)$. See Section 3.2.5.3.

$\mathrm{Th}(i,j)$ the total amount of highlighting that the i-th user performed on the j-th content. This total is calculated through the following equation:

$$\mathrm{Th}(i,j) = \sum_{h=1}^{n} h(i,j)$$

where

h is a highlight performed by the i-th user on the j-th content. The value for each highlighted done is 1.

$\mathrm{TTh}(i,j)$ the total amount of highlights performed by each i-th user on the j-th content. This total is defined as:

$$\mathrm{TTh}(i,j) = \{\mathrm{Th}(1,1), \mathrm{Th}(2,1), \mathrm{Th}(3,1), \ldots, \mathrm{Th}(i,1)\}$$

$\mathrm{SocialValue}(\mathrm{TTt}_k(i,j))$ is the maximum, average or median amount of highlight that the i-th user have performed on the j-th content, within $\mathrm{TTh}(i,j)$. The social value's calculation option is selected depending on the configuration:

$$\mathrm{SocialValue}(\mathrm{TTt}_k(i,j)) = \begin{cases} \mathrm{MAX}(\mathrm{TTt}_k(i,j)) \\ \mathrm{AVERAGE}(\mathrm{TTt}_k(i,j)) \\ \mathrm{MEDIAN}(\mathrm{TTt}_k(i,j)) \end{cases}$$

- **A_4 – Adding notes to a content**

While reading content, the user might want to add its own comments and impressions about it through the notes. This action is usually performed by the users when they read a fragment of the text and want to write down their own thoughts about the content. The value for adding notes to a content it is calculated through an equation system similar to the one used for the A_3 action.

- **A_5 – Commenting a content**

According to the results shown in (Nuñez-Valdéz et al., 2012), when a user comments a content, is because he finds it interesting. Because of this, it is necessary to know if the user has written a comment about the content that is being evaluated. In order to calculate the value of the comments written by a user, we take into account the maximum number of comments written by a user, within the total amount of comments written by all the users on each of the contents of the platform. The value for commenting a content is calculated through an equation system, very similar to the one used for the A_3 action.

- **A_6 – Recommending a content to other contacts**

According to the results shown in (Nuñez-Valdéz et al., 2012), when a user recommends content is because he finds it interesting. In this platform it is necessary to know the amount of recommendations of the content performed by the user in comparison with the recommendations to other contacts performed by all the users of the platform. The value for the recommendation of a content to other contacts is calculated through an equation system, very similar to the one used for the A_3 action.

- **A_7 – Adding a content to the collection**

When a user checks content and adds it to his collection, it might be a sign of interest on that content. The value for adding content to the collection is calculated through the following equation:

$$A_7(i,j) = \begin{cases} \mathrm{Ls}, & \text{if } f = 1 \\ 0, & \text{if } f = 0 \end{cases}$$

where

f is the state of adding a content to a user's collection, where:

$$f = \begin{cases} 1, & \text{if the } j\text{-th content was added to the collection of the } i\text{-th user} \\ 0, & \text{if the } j\text{-th content was not added to the collection of the } i\text{-th user} \end{cases}$$

i is the i-th user that added a content to his collection.
j is the j-th content that was added to the collection of a user.
Ls the superior limit of the value normalization of $A_7(i,j)$.

- **A_8 – Adding a content to the favorites list**

Normally when a user adds a content to his favorites list, it might be a sign of interest on that content. The value for adding a content to the favorites list is calculated through an equation system similar to the one used for the A_7 action.

- **A_9 – Rejecting a content recommendation**

When a contact recommends a content to a user and this one rejects it, it is most likely that he is not interested on it, because the normal thing would be to add it to the collection. The value for rejecting a recommendation is calculated through the following equation:

$$A_9(i,j) = \begin{cases} \text{Li}, & \text{if } f = 1 \\ 0, & \text{if } f = 0 \end{cases}$$

where

f shows the state of rejecting the recommendation of content, where:

$$f = \begin{cases} 1, & \text{if the content } j\text{-th was rejected by } i\text{-th user} \\ 0, & \text{if the content } j\text{-th was not rejected by } i\text{-th user} \end{cases}$$

i is the i-th user that rejected a content.

j the j-th content that was rejected by a user.

Li the inferior limit of the value normalization of $A_9(i,j)$.

- **A_{10} – Eliminating a content from the favorites list**

If a user eliminates a content from his favorites list, the chances for it to be interesting for him depend on certain elements. For example, if a user eliminates content from his favorites list without reading it is probable that he does not like the content, but if the user has read it and still decides to eliminate it the chances could depend on the results obtained through the interaction with that content. The value for eliminating content from the favorites list is calculated through this equation:

$$A_7(i,j) = \begin{cases} \text{Li}, & \text{if } f = 1 \text{ and } A_2(i,j) \leqslant 0 \\ T, & \text{if } f = 1 \text{ and } A_2(i,j) > 0 \\ 0, & \text{if } f = 0 \end{cases}$$

where

i is the i-th user that eliminated a content from his favorites list.

j the j-th content that was eliminated from a user's favorites list.

Li the inferior limit of the value normalization of $A_{10}(i,j)$.

f shows the state of eliminating a content from the collection, where:

$$f = \begin{cases} 1, & \text{if the } j\text{-th content was deleted from collection of the } i\text{-th user} \\ 0, & \text{if the } j\text{-th content was not deleted from collection of the } i\text{-th user} \end{cases}$$

T is the value of the interaction of the i-th user with the j-th content. This value is calculated through the following equation:

$$T = \frac{A_2(i,j) + A_3(i,j) + A_4(i,j)}{N}$$

where $A_2(i,j)$ is the value for the reading time of the j-th content by the i-th user. $A_3(i,j)$ is the value for the highlighting of the j-th content by the i-th user. $A_4(i,j)$ is the value for writing a note in the j-th content by the i-th user.

N is the number of interaction actions with the j-th content performed by the i-th user. This value is calculated through the following equation:

$$N = \sum_{k=2}^{4} f(A_k)$$

where

k is the sub-index that identifies the action.

$f(A_k)$ the function which indicates that the i-th user performed the A_k action on the j-th content.

- **A_{11} – Eliminating a content from the collection**

Just like when eliminating contents from the favorites list, if a user eliminates content from his collection, the chances for it to be interesting for him depend on certain elements, for example, if a user eliminates content from his collection without reading it is probable that he does not like the content, but if the user has read it and still decides to eliminate it the chances could depend on the results obtained through the interaction with that content.

The value for eliminating content from a collection is calculated through an equation very similar to the one used for the A_{10} action.

3.2.5.3. Diagram showing the expected results from the data conversion process through UICA. With the implementation of the mathematical conversion model defined in the previous section, the explicitation algorithm generates a set of data based on a relation "User → Content → Value". In order to obtain optimal results, it is necessary to take into account the following considerations:

- *Define proper normalization limits:* It is necessary to define the proper limits to perform the data normalization in an adequate way. For this article's case study the chosen limits were (1...5) in order to simulate a "five stars" rating system, but any other ranges can be defined, depending on the requirements for the implementation. In this case the values of the variables Li and Ls are 1 and 5, respectively.
- *Choose a calculation method:* To make a comparison between the behavior of one user and the other users of the platform is necessary to define the calculation method that will be used in the process (maximum, mean or median). In the evaluation for this proposal we are doing an analysis using the three calculation methods simultaneously in order to determine which one is the most adequate.
- *Specify the optimal weights:* In order to reach a state of equity between the different actions is necessary to know the level of importance for each action in relation to the rest. This helps to obtain better results when evaluating the different actions. We performed a survey among a group of users about their behavior in electronic books social networks to determinate the optimal weights for the actions defined in our case study. The goal of this survey was to provide rapid evidence of users' behavior in an e-Book social network. The survey sample consisted of 84 users from different ages and sexes via online through email and social networks. The questionnaire requested users to specify the level of importance of certain actions taken on some digital content in a social network. The value assigned to each action corresponds to a scale, where each user specifies how positive or negative is to perform this action in relation to his/her interest in it. In the results obtained from the survey, users consider that the level of importance for all actions were very similar.

4. Evaluation

In this section the proposed architecture is evaluated. To do this, the results obtained from the platform's implementation are analyzed, focusing on the resultant data from the explicitation process of users' behavior. These results show a clear sight of the users' interest on the contents, obtained through the analysis and interpretation of their actions.

The evaluation is done for each user and content within the platform through the comparison of the explicit ratings given by the users to each content and the values generated by the implementation of the explicitation system, through the *User Interactions Converter Algorithm* (*UICA*).

Please cite this article in press as: Núñez-Valdez, E. R., et al. Creating recommendations on electronic books: A collaborative learning implicit approach. *Computers in Human Behavior* (2015), http://dx.doi.org/10.1016/j.chb.2014.10.057

4.1. Methodology

In order to test the proper functioning of the implementation of the proposed architecture and determine how effective the *User Interactions Converter Algorithm* (*UICA*) is when trying to approach a solution such as the explicit feedback, we did a study with a group of users that shared electronic books in "*eInkPlusPlus Social Network*". The users were able to read all the contents and perform the same actions they would be able to perform with a physical book.

These activities were done in an Android device through the application "*eInkPlusPlus Reader*". The test consisted on recommending a number of electronic books to the users (who could accept or reject the recommendations) and then perform the actions shown in Table 1. These actions are the same than they would usually perform with a book in a community for electronic book readers: reading, writing notes, highlighting texts, making recommendations, etc. Once the users had selected the books, they read it and performed the actions they considered appropriate. Then, they had to explicitly rate the books, in order to compare these ratings with the value generated by *UICA* as a result from the analysis and the conversion of the implicit actions performed on the books.

In this study 30 users have participated, interacting with 10 different electronic books. The users (who had been chosen randomly) had different levels of knowledge and different ages (between 16 and 35 year old), and none of them had previous knowledge of the contents. They provided the necessary data for the aforesaid study. As it has been said in previous sections, the data for the study was compiled by the *feedback system* (*UIREC*) defined in the architecture and later converted by the *explicitation system* (*UICA*), which hosted the results that are analyzed below and which will be later exposed for the final conclusions.

4.2. Classification result

The evaluation of the architecture has been done by comparing of the results obtained from the conversion of the implicit data and the explicit data assigned by the users. As it is explained in the definition of the Feedback System, the value of the actions can be calculated by using three calculation methods in order to compare the behavior of a social network user and the rest of the platform's users. These are: the maximum, the average and the median.

With the aim to approach the explicit ratings in the most precise possible way, the explicitation process was performed by separately using the three methods mentioned before. After defining and establishing a series of actions for the users to perform around the electronic books, we must define a conversion mechanism and perform a data conversion process. The obtained results are analyzed and evaluated in this section.

In order to ensure a better comprehension of these results, they are divided into several sections, something that gives us a visualization of them from different perspectives, taking into account important elements, such as the final rating of the content, the resultant value for each action and the social comparison mechanisms used for the conversion. For a clearer vision of the results obtained from the explicitation process, their evaluations are divided into two sections that are defined below and will be explained later:

1. Evaluation of the results from the explicitation by contents outlined by disparity range.
2. Evaluation of the results from the explicitation by contents outlined by disparity.

In addition to present the data taking into account the previous sections, these are analyzed taking the disparity's absolute and real values as a reference. The absolute value helps to adjust the values in order to make the model viable, so the dispersion of data leads to the convergence of the desired value. The analysis of the model with the real values express the dispersion of data in a more realistic and specific way. To make easier to comprehend the data's results with real values, it must be pointed out that the disparity value is equal to the implicit rating minus the explicit rating.

4.2.1. Evaluation of the results from the explicitation by contents outlined by disparity range

As posed in the Mathematical Conversion Model, in order to explicitate the data we must specify a range of values that is equivalent to the limits that define the ratings that a user can give to content. For the implementation of the proposed solution, a [1, 5] range was chosen. This means that the inferior value (1) would be the worst possible rating that the user would give to the content, while the superior value (5) would be the best possible one.

In this section we are doing an analysis in which we group the absolute disparity of the data obtained from the explicitation process against the explicit rating given by the users. The three ranges that have been defined are: [0, 1], [2, 3] and [4, 5], whose results represent an optimal, medium and low approximation, respectively. In addition to these ranges, the evaluation of these results is also done by using the values maximum, average and median. As shown in Fig. 3 the main results obtained in this evaluation are:

- In the First case, the results of explicitation data outlined by disparity range using the median shows that the 75.6% of the explicitation data are optimally close to the expected values, the 15.3% presents a disparity of 2 or 3 levels and only the 9.1% shows a considerable difference in relation to the explicit rating. The approximation of the data explicitation to the explicit rating using UICA shows quite significant results when using the median as a social comparison mechanism.
- In the second case, in a very similar way to the results obtained from the data explicitation using the median, the average as a social comparison mechanism produces very good results in the explicitation process. Observing the results in Fig. 3 it can be seen that the 74.8% of the converted data approximates optimally, with barely a 0.8% of difference in relation to the use of the median. The 17.4% are within an average range of 2 or 3 points of disparity and just a 7.9% of the data are significantly far from the expected values.
- In the latter case, when analysing the results of absolute explicitation using the maximum value, the 61.2% of the explicitated data are optimally near to the expected value, while the 32.6%

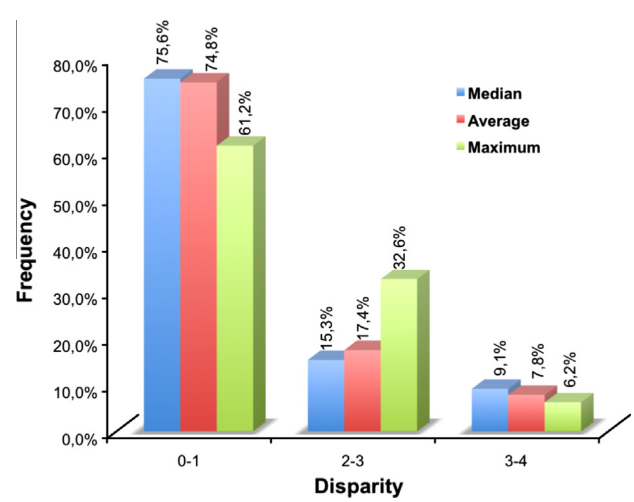

Fig. 3. Histogram of data explicitation results summarized by disparity range.

of the values show a disparity between 2 and 3 points in comparison to the explicit rating and only the 6.2% is far from the expected values, with a difference of between 4 and 5 points. Though the obtained results using the maximum are not as effective as the ones obtained using the average and the median, it can be said that this social comparison mechanism can be valid, because more than the 60% of the data is significantly close to the optimal value.

4.2.2. Evaluation of the results from the explicitation by contents outlined by disparity

In the Section 4.2.1 we analyze the results of the explicitation by dividing the absolute disparity into three ranges and taking into account the social comparison mechanism with which the data was converted, in order to have a more summarized vision of the data. In this section we analyze the data shown in Fig. 4 following the previous structure, but this time taking as a reference model the disparity's absolute value against the model with the real data's values, which express the data dispersion in a more specific way. As shown in Fig. 4, the main results obtained in this evaluation are:

- The first case shows the conversion results obtained using the median. The 25.6% of the results match exactly with the explicit rating given by the users and 50% present a difference of 1 point in relation to the expected value. Only the 6.2% of the data present a difference of 5 points in relation to the expected value. This last difference shows that the algorithm does not match with the results obtained from the user in any of the performed actions.
- In the second case, when analyzing the real values of the explicitation using the statistical median, the 50% of the results have a difference of one point (absolute value's addition) in relation to the expected value, it must be pointed out that in the 16.9% the algorithm calculated 1 point less than the rating given by the user, while 33.1% of the cases assigned one point more. For the remaining cases with differences, the calculated value was higher than the expected one, except for 0.4% that obtained an inferior value of two points.
- In the third case, when using the arithmetic average, the 21.9% of the data matches completely with the value given by the user. If these values are compared with the results generated through the use of the median (first case) it can be seen that the assertion percentage without difference decreases by 3.7%, while the results with 1 and 2 points of difference increase by 2.9% and 2.1%, respectively. On the other hand, the differences of 4 points decrease by 1.2%. Although the results without difference decrease in relation to the median, the general results obtained through the average are satisfactory because the difference percentage between 0 and 1 keeps practically stable, with a 74.8%.
- In the fourth case, Analyzing the real values of explicitation using the arithmetic average as a social comparison mechanisms, where 52.9% of the results have a difference of 1 point

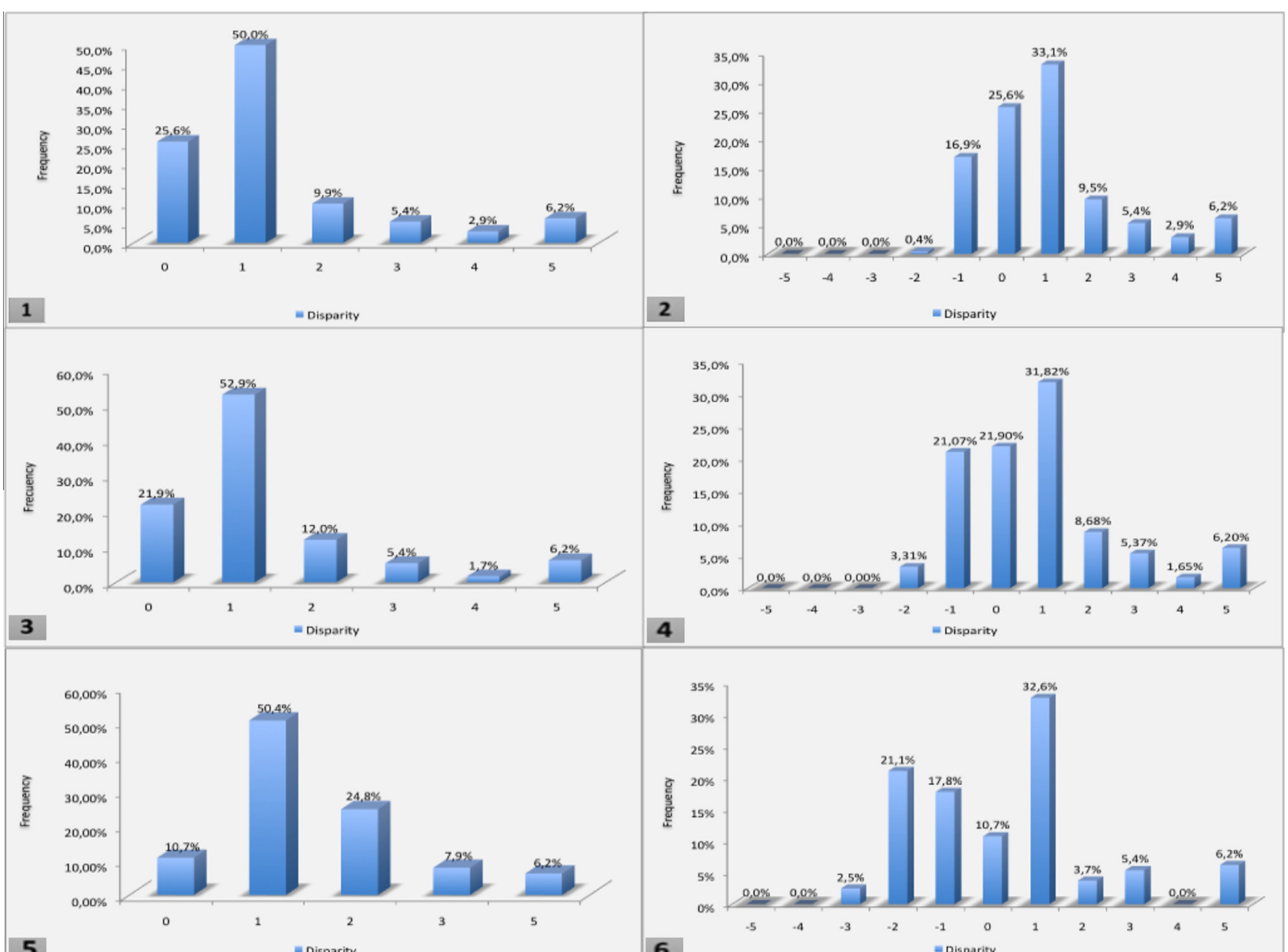

Fig. 4. Histograms of data explicitation results by content.

Please cite this article in press as: Núñez-Valdez, E. R., et al. Creating recommendations on electronic books: A collaborative learning implicit approach. *Computers in Human Behavior* (2015), http://dx.doi.org/10.1016/j.chb.2014.10.057

in relation to the expected value), it can be seen that in 21.1% of the cases, the algorithm calculated 1 point less than the value given by the user, while in 33.8% of the cases, it assigned 1 more point. For the rest of the cases where differences were detected, a value superior to the expected one was calculated, with the exception of 3.3%, where an inferior value of 2 points was obtained. When comparing the results with the ones obtained through the median, it can be seen that they practically follow the same scheme, because the differences increase or decrease in the same data series.

- In the fifth case, when using the maximum value only 10.7% of the data matches completely with the expected value. If we compare this value with the ones obtained through the use of the median or the average (first and third cases) it can be seen that the assertion percentage without difference decreases from more than 50% (25.6% with the median and 21.9% with the average) to just 10.7% using the maximum. However, the difference of 1 point remains practically stable using the three methods (50%, 52.9%, 50.4%). On the other hand, the difference of 2 points increased by more than 50%, practically the same percentage decrease mentioned in the first case.
- In the latter case, Analyzing the real values of explicitation using the maximum value as a social comparison mechanism (where 50.4% of the results has a difference of 1 point in relation to the expected value), it can be pointed out that in 17.8% of the cases, the algorithm calculated one point less than the value given by the user, while in 32.6% of the cases it assigned one more point, values that are very similar to the ones obtained through the median and the average. For the rest of the cases where differences were detected, a value superior to the expected one was calculated, with the exception of 21.1% and 2.5%, where inferior values of 2 and 3 points were obtained, respectively.

5. Discussion

As it has been specified in the study case, the aim of this section is to validate the results obtained from the proposed architecture and, in particular, determine the effectiveness of the *User Interactions Converter Algorithm* (*UICA*) with which the behavior of the users of an electronic book social platform is analyzed and converted into a set of values that are considerably close to the explicit feedback.

Analyzing the evaluation of the explicitation results for the behavior of the users (with which we intend to build more efficient recommendation systems, based on the explicit feedback), we can state the following affirmations, divided into two large groups:

- **General affirmations**

  a. The defined actions (Table 1) significantly represent the most common actions that help to determine the users' interest for the contents of an electronic book platform. In spite of this, other actions can be evaluated and included in the model.
  b. The arithmetical average or the statistical median can be used to compare the behavior of a user in relation to the other users of a platform, because both results tend to be similar and are fairly close to the optimal value.
  c. The maximum value is not advisable for comparing the users' behavior in those actions that can be performed more than once (taking notes, recommending contents, etc), because there are chances for noise factors to appear, which tend to phase out the results. Still, it should not be totally discarded, because the obtained results are fairly decent, with more than 60% of the data matching with a difference range of [0, 1]. Also, it can be illogical to use the maximum in these cases, because it only considers the behavior of one user and discriminating the rest, so it would not constitute a significant sample.
  d. The maximum value can only be used for calculating those actions that commonly are performed once (for example, adding content to the collection), because in this case the average, the median and the maximum would offer the same results.
  e. The analysis and construction define the variables, entities and relations that help to study this complex system, which allows converting implicit feedback into explicit ratings.
  f. The interpretation of the mathematical model through the development and implementation UICA show its accuracy, as the obtained results are pretty closet o the expected value. This is demonstrated by the resulting values, which match in more than 75% with a difference range of [0, 1] in relation to the explicit ratings assigned by the users.

- **Specific affirmations**

  a. When the reading time of a user gets significantly close to the average or the median of the time spent by the other users, there are high chances for him to find the content interesting.
  b. When a user adds a note, highlights, comments or recommends content is because he finds it interesting.
  c. Adding a content to the collection does not average that the users like it, but the trend suggest that if a user adds it to its collection is because he is interested in that content, or at least he was, at a certain moment.
  d. When a user likes content he normally adds it to its favorites list and tends to keep it, not to eliminate it. In this study, none of the users eliminated contents from its favorites list.
  e. When a user rejects a recommendation it becomes evident that he is not interested in that content.

6. Future research directions

Although we have performed a first approach to the use and implementation of recommendation systems based on implicit feedback, this proposal can be refined and extended to bear additional features. Also, it opens different investigation paths to complete and improve the defined methods and tools. Some of these paths are: (1) Implementation of this architecture in other environments, which allow the recommendation of any kind of products. (2) Defining a Domain Specific Language (DSL) that allows to separate the definition of the actions that define the behavior of the users from its implementation.

7. Conclusion

In this paper we have proposed an architecture for the construction of a content recommendation platform based on the behavior of the users of electronic books in the web, aiming to help the users discover contents of their interest automatically and effortlessly.

The goal was to achieve an approximation to a solution for the explicit feedback in the recommendation systems within an environment of electronic books. This architecture allows to analyze the behavior of the user and convert this data into explicit ratings, so they get as close as possible to the ratings that the users would give to the contents in an explicit way. On another hand, this approach allows to build a collective web knowledge in a collaborative learning context.

To verify the feasibility of the proposed solution we have constructed the *ECRP* platform, of which results certify that the proposed architecture allows the development of more effective recommendation systems, based on implicit feedback. Through the results obtained from the implementation of the architecture, we can prove that is possible to determine the users' interest by analyzing and converting their behavior.

As shown in the Evaluation section, 75.6% of the recovered data with the "*User Interactions Recorder* (*UIREC*)" feedback system are optimally close to the expected values. This shows that, with the implementation of this architecture and the use of the applications developed as prototypes in this investigation Project, is possible to develop more efficient recommendation systems that do not depend on the users' explicit ratings.

Lastly, as it has been shown through this paper, the implementation of the processes defined in this architecture is easy to scale and develop. This allows any platform to easily include a recommendation system based on implicit feedback to know the interests of the users and help them to improve their experience and satisfaction.

Acknowledgments

This work was performed by the University of Oviedo under Contract No. TSI-020110-2009-137 of the research project eInkPlusPlus. Project co-financed by the Ministry of Industry, Tourism and Commerce under its National Plan for Scientific Research, Development and Technological Innovation.

References

Adomavicius, G., Sankaranarayanan, R., Sen, S., & Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS), 23*(1), 103–145. http://dx.doi.org/10.1145/1055709.1055714.

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng., 17*(6), 734–749. http://dx.doi.org/10.1109/TKDE.2005.99.

Balabanović, M., & Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. *Communications of the ACM, 40*(3), 66–72. http://dx.doi.org/10.1145/245108.245124.

Claypool, M., Brown, D., Le, P., & Waseda, M. (2001). Inferring user interest. *IEEE Internet Computing, 5*(6), 32–39. http://dx.doi.org/10.1109/4236.968829.

González Crespo, R., Sanjuán Martínez, O., Cueva Lovelle, J. M., Pelayo García-Bustelo, B. C., Gayo, J. E. L., & de Pablo, P. O. (2010). Recommendation System based on user interaction data applied to intelligent electronic books. *Computers in Human Behavior*, In Press. doi:http://dx.doi.org/10.1016/j.chb.2010.09.01.

Kautz, H., Selman, B., & Shah, M. (1997). Referral Web: Combining social networks and collaborative filtering. *Communications of the ACM, 40*(3), 63–65. http://dx.doi.org/10.1145/245108.245123.

Kelly, D., & Belkin, N. J. (2001). Reading time, scrolling and interaction: Exploring implicit sources of user preference for relevance feedback. In *SIGIR 01 Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 408–409). doi:http://dx.doi.org/10.1145/383952.384045.

Kelly, D., & Teevan, J. (2003). Implicit feedback for inferring user preference: A bibliography. *SIGIR Forum, 37*(2), 18–28. http://dx.doi.org/10.1145/959258.959260.

Lee, H. J., & Park, S. J. (2007). MONERS: A news recommender for the mobile web. *Expert Systems with Applications, 32*(1), 143–150. <http://www.sciencedirect.com/science/article/pii/S0957417405003167>.

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE, 7*, 76–80. http://dx.doi.org/10.1109/MIC.2003.1167344.

Montes-García, A., Álvarez-Rodríguez, J. M., Labra-Gayo, J. E., & Martínez-Merino, M. (2013). Towards a journalist-based news recommendation system: The Wesomender approach. *Expert Systems with Applications, 40*(17), 6735–6741. <http://www.sciencedirect.com/science/article/pii/S0957417413004272>.

Noor, S., & Martinez, K. (2009). Using social data as context for making recommendations: An ontology based approach. In *Proceedings of the 1st workshop on context, information and ontologies* (pp. 7:1–7:8). New York, NY, USA: ACM. doi:http://dx.doi.org/10.1145/1552262.1552269.

Nuñez Valdéz, E. R., Cueva Lovelle, J. M., Sanjuán Martínez, O., Montenegro Marín, C. E., & Infante Hernandez, G. (2011). Social voting techniques: A comparison of the methods used for explicit feedback in recommendation systems. *International Journal of Interactive Multimedia and Artificial Intelligence, I*, 61–66.

Nuñez-Valdéz, E. R., Cueva Lovelle, J. M., Sanjuán Martínez, O., García-Díaz, V., ordoñez de Pablos, P., & Montenegro Marín, C. E. (2012). Implicit feedback techniques on recommender systems applied to electronic books. *Computers in Human Behavior*.

O'Donovan, J., & Smyth, B. (2005). Trust in recommender systems. In *Proceedings of the 10th international conference on intelligent user interfaces* (pp. 167–174). New York, NY, USA: ACM. doi:http://dx.doi.org/10.1145/1040830.1040870.

Papagelis, M., Plexousakis, D., & Kutsuras, T. (2005). Alleviating the sparsity problem of collaborative filtering using trust inferences. In *Proceedings of the third international conference on trust management* (pp. 224–239). Berlin, Heidelberg: Springer-Verlag. doi:http://dx.doi.org/10.1007/11429760_16.

Pommeranz, A., Broekens, J., Wiggers, P., Brinkman, W.-P., & Jonker, C. M. (2012). Designing interfaces for explicit preference elicitation: A user-centered investigation of preference representation and elicitation process. *User Modeling and User-Adapted Interaction, 22*(22), 357–397. http://dx.doi.org/10.1007/s11257-011-9116-6.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on computer supported cooperative work* (pp. 175–186). New York, NY, USA: ACM. doi:http://dx.doi.org/10.1145/192844.192905.

Resnick, P., & Varian, H. R. (1997). Recommender systems. *Communications of the ACM, 40*(3), 56–58. http://dx.doi.org/10.1145/245108.245121.

Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 253–260). New York, NY, USA: ACM. doi: http://dx.doi.org/10.1145/564376.564421.

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence, 2009*(Section 3), 1–19. http://dx.doi.org/10.1155/2009/421425.

Taghipour, N., Kardan, A. (2008). A hybrid web recommender system based on Q-learning. In *Proceedings of the 2008 ACM symposium on applied computing* (pp. 1164–1168). New York, NY, USA: ACM. doi:http://dx.doi.org/10.1145/1363686.1363954

Terveen, L., Hill, W., Amento, B., McDonald, D., & Creter, J. (1997). PHOAKS: A system for sharing recommendations. *Communications of the ACM, 40*(3), 59–62. http://dx.doi.org/10.1145/245108.245122.

Verbert, K., Manouselis, N., Ochoa, X., Wolpers, M., Drachsler, H., Bosnic, I., et al. (2012). Context-aware recommender systems for learning: A survey and future challenges. *IEEE Transactions on Learning Technologies, 6*(1), 2007. <http://dspace.ou.nl/handle/1820/3855>.

Zhang, Z.-K., Zhou, T., & Zhang, Y.-C. (2011). Tag-aware recommender systems: A state-of-the-art survey. *Journal of Computer Science and Technology, 26*(5), 767–777. http://dx.doi.org/10.1007/s11390-011-0176-1.

Ziegler, C. -N., McNee, S. M., Konstan, J. A., & Lausen, G. (2005). Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web* (pp. 22–32). New York, NY, USA: ACM. doi: http://dx.doi.org/10.1145/1060745.1060754