JISTEM: Journal of Information Systems and Technology Management
E-ISSN: 1807-1775
tecsi@usp.br
Universidade de São Paulo
Brasil

Valois B. Jr, Cleomar; Armada de Oliveira, Marcius
RECOMMENDER SYSTEMS IN SOCIAL NETWORKS
JISTEM: Journal of Information Systems and Technology Management, vol. 8, núm. 3, septiembre-diciembre, 2011, pp. 681-716
Universidade de São Paulo
São Paulo, Brasil

JISTEM - Journal of Information Systems and Technology Management
Revista de Gestão da Tecnologia e Sistemas de Informação
Vol. 8, No. 3, Sept/Dec. 2011, *pp. 681-716*
ISSN online: 1807-1775
DOI: 10.4301/S1807-1775201100030009

RECOMMENDER SYSTEMS IN SOCIAL NETWORKS
SISTEMAS DE RECOMENDAÇÃO EM REDES SOCIAIS

Cleomar Valois B. Jr
Marcius Armada de Oliveira
Centro Universitário da Cidade do Rio de Janeiro, RJ - Brazil

ABSTRACT

The continued and diversified growth of social networks has changed the way in which users interact with them. With these changes, what once was limited to social contact is now used for exchanging ideas and opinions, creating the need for new features. Users have so much information at their fingertips that they are unable to process it by themselves; hence, the need to develop new tools. Recommender systems were developed to address this need and many techniques were used for different approaches to the problem. To make relevant recommendations, these systems use large sets of data, not taking the social network of the user into consideration. Developing a recommender system that takes into account the social network of the user is another way of tackling the problem. The purpose of this project is to use the theory of six degrees of separation (Watts 2003) amongst users of a social network to enhance existing recommender systems.

Keywords: Recommender Systems, Social Networks.

1. INTRODUCTION

1.1. Position and Explanation

In the past, the study of knowledge discovery in databases, and more specifically of recommender systems (systems that filter relevant information to a specific user according to his/her profile), was limited to the data available to researchers. With the Internet explosion, new opportunities and challenges have arisen in this research field.

More recently, the online social network phenomenon has enabled access to the user's profile and preferences. This has allowed several databases available on the Internet to be collected and used in experiments by research (GroupLens Research 2010).

The importance of research on recommender systems arises from the wide scope of the available information, making it difficult for the user to access relevant items and raising the need for tools that help perform this task.

Manuscript first received/*Recebido em* 05/03/2011 Manuscript accepted/*Aprovado em*: 17/11/2011

Address for correspondence / *Endereço para correspondência*

Cleomar Valois Batista Jr., Bachelor of Science in Computer Science (BSc CS), Centro Universitário da Cidade do Rio de Janeiro, NUPAC - Núcleo de Projetos e Pesquisa em Aplicações Computacionais, Rio de Janeiro, RJ - Brazil, +55(21)3593-6186, E-mail:jr@muitobom.com.br

Marcius Armada de Oliveira, Bachelor of Science in Computer Science (BSc CS), Centro Universitário

For this same reason, the generation of good recommendations has been commercially exploited by large companies. Although these recommendations are acceptable, they still present several problems. The data used often does not correctly reflect the client's preferences. The items purchased are not always intended for the buyers themselves. They may be gifts, and this fact makes the data unreliable (Gupta, Jain e Song 2008).

Most recommender systems researched and in operation make recommendations without taking into consideration any knowledge about the recommended items. For that reason, they must take into account all data space known by the system.

However, because of the easy access to large quantities of information, usually beyond processing capacity in a reasonable response time, recommender systems not only need to deal with the quality of the recommendations, but they also must filter the available base with which they are going to work.

In order to verify a possible solution to this scalability problem (section 2.4.4.7) and reduce the volume of information, this article intends to use the concept that people closer to a person in a social network have more influence on his/her opinions (Mendes 2008). This way, the existing data space may be divided based on the degree of separation among the individuals within the social network and still generate relevant recommendations in a database that would otherwise not be processed due to its extent.

1.2. Objective

Recommender systems are widely used in several different domains for the recommendation of articles, music, movies, and even people. Portals such as Amazon and Submarino use recommender systems to suggest products to their customers. Meanwhile, social networks such as LinkedIn and Facebook use them to suggest new contacts.

To accomplish that, the most used techniques employed in recommender systems are (section 2.4.3): The collaborative filtering and content-based systems. The collaborative filtering does not take into account the type of items, nor their attributes. It takes exclusively into account the expressed opinion about the other items in order to make recommendations. Meanwhile, content-based filtering uses the knowledge it has of the items and their attributes to make recommendations.

These techniques perform well, but they employ cluster[1] solutions to solve the scalability problem (section 2.4.4.7) and be able to process high chunks of data.

This article looks for a new solution, different from the one normally employed. The objectives of this article are:

- From an assessment database, similar to Grouplens' (GroupLens Research 2010) initiative, complemented with the relations among the participants, in such a way that is possible to draw the graph of the social network;
- Evaluate the benefits to recommender systems originated from the knowledge provided by the social network. This database has missing

[1] A cluster is composed by a group of linked computers that uses a special type of operational system, called distributed system. It is often built from traditional computers (PCs) connected to a network. They communicate with each other through the system, operating as a single big machine.

values that are treated as non-evaluated items. Recommender systems try to predict the user's evaluation of an item that has not yet been evaluated. Based on the concept that people who are more closely connected have more influence on each other's opinions (Mendes 2008), this article will try to predict these missing values based on relations, so as to generate more relevant results.

- Evaluate the applicability and imputation efficiency of the missing evaluations with refeeds (section 2.3), using the identified degrees of separation among participants. Verify, in contrast to the traditional approach, if there are scenarios in which this application is more useful.

At the end of this article, results will be presented which indicate whether dividing the data base by degrees of separation (section 2.2) will have a positive effect on the results.

2. FOUNDATIONS AND TECHNOLOGIES

2.1. Social Networks

Social networks are node structures (individuals or organizations) connected by social ties. The organization of each network depends on the surroundings in which it was generated and operates. Each network has a particular organization of its members and, especially, of its facilitators' political culture and shared objectives (Amaral 2004). The social network represents a set of independent members joining ideas and resources based on shared values and interests (Marteleto 2001).

It is important to understand the classification of networks, as well as how they are formed. In "The Small World" model, ties are established randomly and some people are able to transform the network into a small world. In the book "Six Degrees of Separation – Small World", Watts (Watts 2003) discusses the idea that the average distance between two people on the planet does not exceed approximately 6 people, considering there are some random ties among groups. This topic will be discussed in more detail in section 2.2.

2.2. Six degrees of separation theory

Six degrees of separation is related to the idea that one person is only six "steps" away from any other person on Earth. So, in a social network, in order to connect two people, six or fewer intermediaries would be needed, that is, it is believed that two individuals can be connected through a maximum of 5 acquaintances (Figure 2.1 Six Degrees of Separation

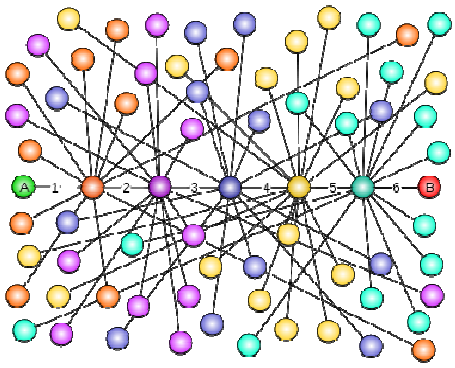


Figure 2.1 Six Degrees of Separation

Source: (Watts 2003)

In his doctorate thesis, Michael Gurevich (Gurevich 1961) produced a seminal work in his empirical studies of social networks structure. The Austrian mathematician Kochen Manfred, involved in an urban statistics project, extended these empirical results to a mathematic manuscript (Sola Pool and Kochen 1978) and concluded that in a population as big as the USA, without social structure, *"it's virtually correct to say that two individuals can get in touch with each other through at least two intermediaries. In a socially structured population this is less probable, but it still can happen, and maybe if we consider the whole world population, probably only one more transition individual is needed"*.

Simulations performed in 1973, using relatively limited computers, produced more realistic predictions of 3 degrees of separation among the US population, anticipating the results of the American psychologist Stanley Milgram, whose study (Milgram 1967) showed that "people in the US seem to be connected on average by a chain of three acquaintances. The simulations did not, however, speculate on the planet interactions".

In 2001, Duncan Watts (Watts 2003), a Columbia University professor tried to recreate Milgram's experiment on the internet, using an email message as a "package" to be delivered. The experiment involved 48,000 senders and 19 intended recipients (in 157 countries). Watts found the average number of intermediaries remained around 6. Note that this was not the highest number of intermediaries.

2.3. Sequential Imputation

Imputation is any automatic or semi-automatic procedure capable of filling in missing values in data bases (Goldschmidt and Passos 2005). Imputation methods, once restricted to the statistics domain, are now more evolved and present implementations based on Artificial Intelligence (IA) or even in hybrid constructions (Farhangfar, Kurgan and Pedrycz 2007) (Lakshminarayan, Harp, and Samad 1999).

Imputation processes can be distinguished by the capacity of imputing missing values that occurs in a univariate or multivariate way. The imputation is univariate when the missing values are present in only one attribute. In this scenario, several techniques are commonly applied, such as:

- Replacement with a central trend value: The missing data of quantitative variables are replaced with the variable's mean. This mean may be the average of the observed data or the average of a group with more similar characteristics to the missing data, identified by one or more categorical variables found in the database.
- "Hot deck": The individual whose observed data most closely resembles that of the individual with the missing data in relation to the auxiliary variables is located. The missing data is then replaced with the corresponding matched data.
- Regression: The imputed values are predicted through regression by simply using one or more existing variables to predict the missing value of another variable similar to the previous one. Two types of regression may be used for the imputation: regression and regression with an additive component of the error variance.

However, due to the increased complexity and database sizes in past years, there has been an intensification of research and development related to the mechanisms for multivariate imputation, when missing values are arranged in two or more attributes (Schafer 1997).

There are two ways of approaching the solution for multivariate problems (Vanbuuren, et al. 2006): joint modeling or fully conditional specification. Joint-modeling consists of using statistical models (like Bayesian network) to stimulate missing values in all attributes at once.

Conditional specification techniques are generalized in literature as sequential imputation (Commission and Europe 2000), in which the missing values to be imputed are processed in a sequential manner based on attributes (Lepkowski, et al. 2001) (Oudshoorn, Buuren, and Rijckevorsel 1999) or records (Kim, Kim e Yi 2004) (Verboven, Branden e Goos 2007). The main characteristic of imputing the missing values sequentially is the breakdown of a multivariate problem into several univariate problems, which can be solved by the well-known traditional technique of univariate imputation. (Oudshoorn, Buuren, and Rijckevorsel 1999) (Gelman and Hill 2006).

2.4. Recommender Systems

2.4.1. Introduction

The explosive growth of the World Wide Web (www), the emerging popularity of e-commerce and social networks have provided access to a large quantity of information, which was previously inaccessible. Gathering data is not a problem anymore, but the extraction of useful information and its presentation to the user in a relevant way is. Recommender systems have been developed to help fill the gap between information collection and analysis, by filtering all available information and presenting the most relevant items to the user (Resnick e Varian 1997). The recommender system helps enhance the capacity and efficiency of this process. The biggest challenge of this type of system is finding the perfect match between those

recommending and those receiving the recommendation; that is, defining and discovering the relation between their interests.

Information systems that filter relevant information for a specific user based on his/her profile are known as Recommender Systems. A recommender system usually compares the user profile with some reference characteristic and attempts to predict the evaluation a user would provide of a particular item that has not yet been considered. E-commerce websites are currently the main interest group of recommender system usage, employing different techniques to find more appropriate products for their clients and to raise sales volume.

There are two approaches to recommender systems: Collaborative filtering and content-based systems, which will be explained in details in section 2.4.3. Some authors, like Montaner (Montaner, Lopez, and de La Rosa 2003), highlight the existence of a third type of information filtering, called demographic filtering. This filtering method is not part of the scope of this article. However, this method uses a person's description to determine the relation between a specific item and the type of individual who may be interested in the particular item. This kind of approach uses peoples' descriptions to determine the relations between one item and the type of person who may be interested in that particular item. The user profile is created by grouping users in stereotype classifications that represent the characteristics of a class of users. Personal data are usually requested from the user through registration forms used to create a characterization of the user and his/her interests.

One of the algorithms commonly used in recommender systems is K-NN (section 2.4.5.1). In a social network we may find neighbors of a specific user with the same tastes or interests. In order to do this, the Pearson Correlation coefficient must be calculated through the selection of top-N neighbors preferred data of a specific user (weighted similarity) and use specific techniques to calculate whether the user's preference can be predicted.

2.4.2. Evaluations

The users may make explicit or implicit evaluations. Explicit evaluations are usually a discrete value that belongs to a limited set of possible numerical values for each item (Resnick e Varian 1997), like a Likert type scale. Implicit evaluations offer the advantage of reducing the user workload and are usually extracted from the buying history or the behavior while browsing through sites that require some type of evaluation.

2.4.2.1. Likert SCALE

On a Likert-type scale (Table 2.1 Likert scales), the answers to each item vary according to an intensity degree. This scale with sorted, equally spaced categories and the same number of categories in all items is widely used. It is formed by a set of phrases or sentences with positive or negative opinions. The evaluator (user) has to rate his/her degree of agreement, from "I strongly disagree" (level 1) to "I totally agree"

(level 5, 7 or 11), depending on the number of levels on the scale[2] (Chisnall 1973) (Likert 1932).

Table 2.1 Likert scales

Value	Graphic representation	Textual representation
5	☆ ☆ ☆ ☆ ☆	Excellent
4	☆ ☆ ☆ ☆	Very good
3	☆ ☆ ☆	Good
2	☆ ☆	Fair
1	☆	Poor

2.4.3. Collaborative filtering and content-based filtering

(Balabanovic and Shoham 1997) defined two main approaches to recommender systems: collaborative filtering and content-based filtering. For the collaborative filtering, the recommendation is based on the analysis of similar users to indicate items of preference. For content-based filtering, the recommendation is made through the analysis of items which are similar to the ones the user has already seen and evaluated.

2.4.3.1.Collaborative Filtering

In this type of filtering, recommendations are made based on predictions of user preferences resulting in interactions between other users. This type of filtering usually offers a higher degree of surprise to the user with good recommendations and, in some cases, may offer totally irrelevant contents. Collaborative filtering systems are trying to include people in the filtering systems, since they can better assess documents than any computer task (Resnick e Varian 1997).

[2] Likert previously recommended a scale of 5 points, but is currently recommending the use of scales of 5, 7 or 11 points based on individual's lack of discriminatory capabilities when the scale has many answer possibilities, or based on the fact that only when the scale has many possibilities is it similar to the *continuum* of our opinion.

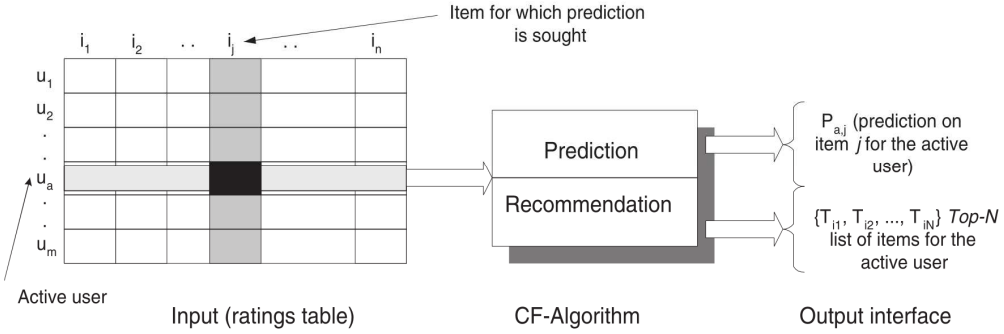


Figure 2.2 Collaborative Filtering

Source: (Sarwar, et al. 2001)

A first approach to this type of filtering (Resnick e Varian 1997) establishes recommendations based on items consumed by users with the same consumption pattern as the current user (Figure 2.2 Collaborative Filtering

It is used mainly in e-commerce systems, such as Amazon and Submarino. To make this type of approach easier to understand, let's consider x a user and X_i a set consisting of the top i users whose buying pattern is more similar to x's (since they bought some of the same products x has brought). Now consider vectors (p,n), where p is a product and n is the number of times this product was purchased by a X_i. If the vector set (p,n) is sorted in s descending order by n, the result will be an order for product recommendation for x. A variation may apply different weights to users X_i, based on their relation to x users.

In the approach related to collaborative filtering, it is possible to solve the problem found in the recommendation approach using contents (section 2.4.3.2), where the user only receives items with similar contents. However, this approach does not solve other problems, such as the insertion of new items into the base that will only be recommended after a certain number of users have read and assessed them. Another issue is handling users who do not have similar interests in other members of the population. Thus, this unique user will not have peers in which the collaborative recommender system can base itself on. This and other issues are described in section 2.4.4.

2.4.3.2.Content-Based Filtering

The content-based filtering approach (Figure 2.3 Content-based Filtering) is based on the premise that the user would like to see similar items as to those previously seen by him (Balabanovic e Shoham 1997). With information on a specific content and data about a specific user that can be related to this information, it is possible to define the relation between user and content. This approach employs content-based filtering techniques, for example, filtering by keyword and latent semantic analysis.

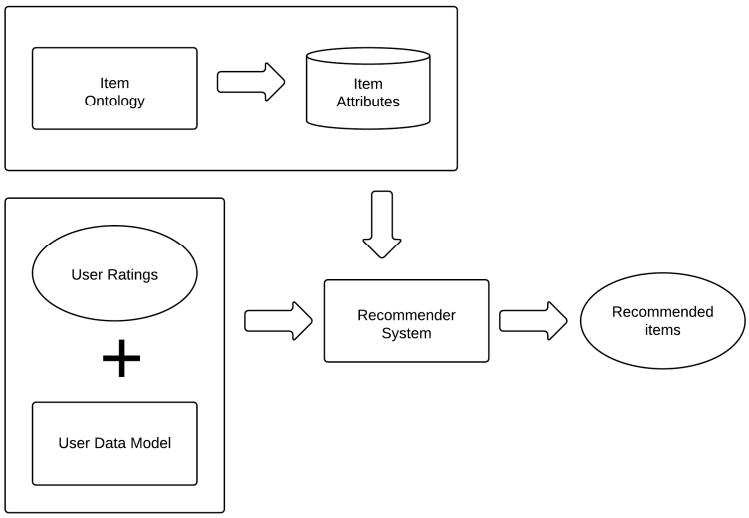


Figure 2.3 Content-based Filtering

Source: (Lorenzi 2006)

For example, a book about programming could be recommended to students of computer related courses. This is the basis of content-based recommendations which, in contrast to collaborative filtering, does not use the relations among users to define the content. For this reason, the recommendation based on contents usually does not surprise the user, since the relation and the means used by the system to recommend can be inferred directly by the user, even if unconsciously.

2.4.4. RECOMMENDER SYSTEMS ISSUES

2.4.4.1.Cold Start

The issue with *Cold Start* is more prevalent in recommender systems. The problem occurs at the start of the system (no assessments from other users). A recommender system usually compares the user profile with some reference characteristics. These characteristics can be based on information (content-based approach) or on the user's social environment (collaborative filtering approach). In content-based approach, the system should be able to match the characteristics of an item to relevant characteristics in the user's profile. In order to do that, a model with sufficient detailed information on the user, including his/her tastes and preferences must first be built. This can be done in explicitly (by consulting the user) or implicitly (observing the user's behavior). In both cases, the *Cold Start* issue requires the user to dedicate to creating his/her profile before the system can begin any relevant recommendation. Because of the *Cold Start* issue, items not previously assessed would be ignored in the collaborative filtering approach.

2.4.4.2. Gray Sheep

If a user has rare tastes, the recommendation may not be accurate, as there are no "close neighbors". This problem is called gray sheep (Resnick and Varian 1997) (Lorenzi 2006). In the collaborative filtering system, a user with this profile is not easily related to other users in the system, making it difficult to recommend items. In the content-based filtering system, even if the user has a rare profile, the recommendation of items related to this profile is not an issue, since recommendations are more generic. For example, if the system identifies that a user is interested in technology and oceanography, it will easily recommend these items to the user, even if only unpopular items have been evaluated.

2.4.4.3.Early-Rater

When a new item emerges, it cannot be recommended to a user before a person assesses it (Resnick and Varian 1997) (Lorenzi 2006). This issue is clearly identified in collaborative filtering. When a new item with no user assessment or recommendations is inserted, it cannot be recommended. In content-based filtering, knowing the contents of an item is enough to enable a recommendation to a user.

2.4.4.4.SPARSE EVALUATIONS

When there are few users and many items, the evaluations may become sparse and it becomes difficult to find similar users (Resnick e Varian 1997) (Lorenzi 2006). In collaborative filtering, this issue is easily identified because the filtering is completely based on the user's assessment of the item. In content-based filtering, the recommendation does not depend on the number of users and items, but rather on their profiles and contents.

2.4.4.5.Super-SPECIALIZATION

Only items that are similar to those previously evaluated by the user will be recommended. Exploring new item categories is not possible (Resnick and Varian 1997) (Lorenzi 2006). In content-based filtering, this issue is clearly identified. A user whose profile has been defined will always receive items related to this profile, and any personal profile modification (outside the system) will not be reflected on the system. In collaborative filtering, item recommendation is not based on the user's initial profile, but rather on his/her actions and relation to other users.

2.4.4.6.Serendipity

This is related to the lack of surprise in the recommendation. Products that are not related to the user's profile may never be recommended (Resnick and Varian 1997) (Lorenzi 2006). This problem occurs in content-based filtering, since the recommended contents will always belong to the same group relating back to the user profile. Meanwhile, in collaborative filtering, the surprise occurs more frequently, since similar users may have evaluated completely different items from those seen by the original user.

2.4.4.7.Scalability

When the quantity of users, items and evaluations is too large, the system that executes real-time calculations of the relations among users may provide a very long response time and may need computer resources that are not available. This is a common problem in both approaches. However, in collaborative filtering, this issue is more evident as the calculations are done using all users and all items. In content-based filtering, calculations are done using only one user and all related items, considering all attributes (Resnick and Varian 1997) (Kajimoto, et al. 2007).

2.4.5. RECOMMENDER SYSTEM TECHNIQUES

The objective of recommender systems is to provide recommendations based on recorded information on the users' preferences. These systems use information filtering techniques to process information and provide the user with potentially more relevant items. This section presents collaborative filtering techniques based on users and items (J., et al. 1999) (Sarwar, et al. 2001).

For **K** users and **M** items, users' evaluations are represented in the **K** x **M** **matrix**, user-item (Figure 2.4 User x Item Matrix
). Each element $x_{k,m} = r$ indicates that user **k** evaluated item **m** with **r**, where **r** $\in \{1, \dots, |r|\}$, that is, this item has been evaluated. And $x_{k,m} = 0$ indicates that the evaluation is unknown.

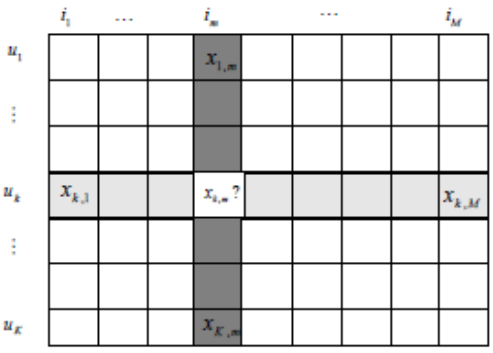


Figure 2.4 User x Item Matrix

Source (Wang, P. and J.T. 2005)

User-item matrix (Figure 2.4 User x Item Matrix
) may be decomposed into row vectors:

$$X = [u_1, \dots, u_k]^T, u_k = [x_{k,1}, \dots, x_{k,M}]^T, k = 1, \dots, K$$

Where T stands for transposition. Each row vector u_k^T corresponds to a user profile and represents a particular item evaluation. This decomposition leads us to collaborative filtering based on users.

Alternatively, the matrix may also be represented by column vectors:

$$X = [i_1, \dots, i_k], i_m = [x_{1,m}, \dots, i_{k,M}]^T, m = 1, \dots, M$$

Where each column vector i_m corresponds to a specific item, which has been evaluated by all **K** users. This representation results in the item-based recommendation algorithm.

2.4.5.1.Based On users (k-NN)

The closest k-neighbor algorithm (k-nearest neighbors), originally proposed in (Cover and P. E. 1974), is based on the similarity concept to build a group of objects (the closest users) from where candidates are extracted in order to impute the evaluation of an item. The similarity concept is based on the idea of distance. Perhaps the Euclidean distance is the most preferred in literature because it was the first to be proposed or perhaps because of its calculation simplicity (Ferlin 2008).

Collaborative filtering based on users predicts user interest in an item based on evaluations of similar users (J., D. and C. 1998) (J., et al. 1999). As shown in Figure 2.5 Evaluation prediction based on user similarities , each user profile is classified based on its dissimilarity to the user profile, for which the prediction is being made. The evaluations performed by the most similar users have more influence on the prediction of the item evaluation for the relevant user. The list of the most similar users may be identified by using a cut-factor or by selecting the most similar *top-N* users.

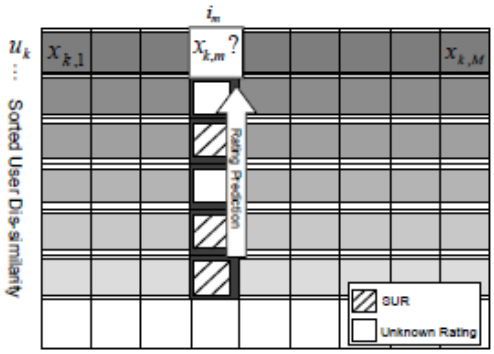


Figure 2.5 Evaluation prediction based on user similarities

Source (Wang, P. and J.T. 2005)

The Euclidean distance and the Pearson correlation are common similarity measures in collaborative filtering (P., et al. 1994) and will be discussed in sections 2.4.6.1 and 2.4.6.2, respectively. The existing methods differ in the way they handle unknown evaluations. Unknown evaluations may be interpreted as a zero value evaluation (P., et al. 1994), or by interpolation[3] of the mean of the user's evaluations and the mean of the evaluations of similar users' evaluations (Xue, et al. 2005).

After defining the most similar users, using only known evaluations of these users, the evaluation of the specific user is estimated (P., et al. 1994).

[3] Through interpolation, we can create a function that somewhat "fits" this specific data, thus giving them the desired continuity.

The user based recommendation algorithm (Return top-N items from list A, sorted by the average of the evaluations in descending order
) can be described as the process of recommending items for a user this way:

---

for each user v (that is not u)

    compute the similarities between u and v

    add the most similar users to the list L of u "neighbors"

for each item i evaluated by a user in L but not evaluated by u

    for each user v in L that has evaluated i

        compute the similarity s between u and v

        Multiply the evaluation of v for item i by weight s ($a_{v,i} = a_{v,i} * s$)

        incorporate the weighted evaluation into the mean of the evaluations for item i

    add the mean of the evaluations of item i to list A

Return top-N items from list A, sorted by the average of the evaluations in descending order

---

Algorithm 1 Recommendation algorithm based on user

Source: (Owen, et al. 2010)

First of all, the most similar users are identified in order to know which items they find interesting. These items are considered as candidates for recommendation to intended users.

2.4.5.2. Slope-One

This approach pre-computes the average difference between the evaluations of each pair of items previously evaluated by the user. The Slope-One system is being recommended as the new reference system for recommender systems in production by (Lemire e Maclachlan 2005) for the following reasons.

- Supports dynamic updates: The addition of new evaluations to the system changes all predictions instantaneously;
- Efficient when consulted: The searches are fast, even though the system requires larger storage capacity than other approaches;
- A user with few evaluations should receive relevant recommendations;
- Reasonably precise: Several approaches "compete" for the most accurate prediction. However, even the smallest gain in precision is not always worth the sacrifice in simplicity or scalability.

As an example, let's suppose that people who enjoyed the movie "Carlito's Way" apparently also liked another movie starred by Al Pacino, "Scarface". However they seem to like "Scarface" more. Let's suppose that on a five star scale (section 2.4.2.1), most people who watched "Carlito's way" gave the movie 4 starts and "Scarface" 5 stars. According to this reasoning, if another person gave "Carlito's Way" 3 stars, it would be possible to assume that this same person would give "Scarface" 4 starts, one more star. Furthermore, people who evaluated "Scarface" gave "The

Godfather" the same rating. Subsequently, a user who evaluated "Carlito's Way" with 2 stars and "The Godfather" with 4 stars would have his/her evaluation estimation for "Scarface" calculated the following way: Based on the evaluation of "Carlito's Way", 2.0 + 1.0 = 3.0. Based on the evaluation of "The Godfather", 4,0 + 0,0 = 4,0. By calculating a simple average of these two evaluations, the estimated evaluation equals (3,0+4,0)/2=3,5. This is the essence of the Slope-One approach.

The name Slope-One originates from the fact that the recommendation algorithm is based on the assumption that there is a linear relation between the evaluation values of two items, and that it is usually possible to estimate the evaluations of an item Y based on the evaluations of an item X, using a linear function similar to Y = mX + b. Slope-One further simplifies this assumption by applying value 1 to m. Therefore, simply find value b = Y – X, and calculate the average difference of the evaluation value for each pair of items.

This represents a significant preprocessing phase (add the average difference $d_{i,j}$, to list D), in which all differences are computed:

---

for each item i

  for each item j (not i)

    for each user u that evaluated i and j

      adds the difference ($b = a_{u,i} – a_{u,j}$) to an average

    add the average difference $d_{i,j}$, to list D

Algorithm 2 Slope-One pre-processing

Source: (Owen, et al. 2010)

After the preprocessing phase, the recommender system that uses Slope-One is able to make recommendations.

---

for each item i not evaluated by user u

  for each item j not evaluated by user u

    find the average difference between i and j in list D

    adds this difference to u's evaluation of j ($a_{u,j} + d_{i,j}$)

    adds this value to an average

returns top-N items sorted by these averages

Algorithm 3 Slope-One processing

Source: (Owen, et al. 2010)

Slope-One performance does not depend on the number of users in the matrix **K** x **M** (Figure 2.4 User x Item Matrix). It depends exclusively on the average difference between every pair of items, which can be pre-computed. Further, this structure can be efficiently updated. Simply update the average difference whenever there is a new evaluation or a change in an existing preference. (Owen, et al. 2010).

However, please note that the memory requirements necessary to store all these differences grow exponentially according to the number of items (items^2) (Owen, et al. 2010).

2.4.6. Similarity metrics

Similarity metrics are functions that numerically measure the similarity degree between two entities. As far as recommender systems are concerned: items or users. Metrics are usually needed in recommender systems in which entities are compared one by one, and the similarity measure is the instrument used to distinguish, among entities, the similar and non-similar candidates (Webber 1998).

The similarity metrics summarize, by a measure of importance, the similarity of each attribute level and are used to model the relevance among them. Consequently, any math-oriented calculation is valid, such as the weighted average. However, it is still possible to perceive the similarity evaluation as a problem of recognizing standards within the entity space (Cover and P. E. 1974).

2.4.6.1.EUCLIDEAN DISTANCE

In mathematics, Euclidean distance (or metric distance) is the distance between two points, which can be proved by the repeated application of the Pythagorean Theorem. If this formula is applied as distance, the Euclidean space becomes a metric space.

The idea makes sense for recommender systems, if the users were represented as points in a space with several dimensions, one for each item, where the coordinates are the evaluation value. This metric computes the Euclidean distance between these two points (users). This value by itself is not a valid similarity metrics because higher values would mean bigger distance and lower level of similarities. The values have to be lower when the users are more similar. For that, in a system implementation, the similarity measure is calculated this way (Table 2.2 Euclidean distance and similarity measure calculated in relation to user 1): 1 / (1 + d), that is, when the distance equals to zero (meaning identical evaluations by both users), the calculated similarity measure is 1, lowering up to zero, according to distance increase (Owen, et al. 2010).

Table 2.2 Euclidean distance and similarity measure calculated in relation to user 1

	Item 101	**Item 102**	**Item 103**	Distance	Similarity related to user 1
User 1	5.0	3.0	2.5	0.000	1.000
User 2	2.0	2.5	5.0	3.937	0.203
User 3	2.5	-	-	2.500	0.286
User 4	5.0	-	3.0	0.500	0.667
User 5	4.0	3.0	2.0	1.118	0.472

source: (Owen, et al. 2010)

The formula for calculating the Euclidean distance is presented below:

$d(a,b) = \sqrt[2]{\sum_{i=1}^{n}(a_i - b_i)^2} = \|a - b\|_2$, considering a and b as data vectors.

2.4.6.2.The Pearson CORRELATION COEFFICIENT

The Pearson correlation coefficient is a measure of the degree of linear relation between two variables. This coefficient varies between -1 and 1. Zero (0) value means that there is no linear relation; value 1 indicates a perfect linear relation; and value -1 also indicates another perfect linear relation, but reverse, that is, when one of the variables increases the other decreases. The closest to 1 or -1, the strongest is the linear association between the two variables.

The calculation formula of Pearson correlation coefficient ρ is:

$$\rho = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(\Sigma(x_i - \bar{x})^2)(\Sigma(y_i - \bar{y})^2)}}$$

$\rho = 1$ It means a perfect positive correlation between the two variables.

$\rho = -1$ It means a perfect negative correlation between the two variables – that is, if one increases, the other always decreases.

$\rho = 0$ In terms of linearity, it means both variables do not depend on each other. However, another kind of dependency (non-linear) may exist. Thus, the result ρ = 0 must be investigated by other means.

The Pearson correlation coefficient measures the trend of two series of numbers, paired one by one, in moving together (Owen, et al. 2010).

Table 2.3 The Pearson correlation related to user1

	Item 101	Item 102	Item 103	Correlation with user 1
User 1	5.0	3.0	2.5	1.000
User 2	2.0	2.5	5.0	-0.764
User 3	2.5	-	-	-
User 4	5.0	-	3.0	1.000
User 5	4.0	3.0	2.0	0.945

source: (Owen, et al. 2010)

2.4.6.3.Log-Likelihood

Log-Likelihood similarity metrics is similar to the Tanimoto Coefficient, since evaluations done by the users are not taken into consideration, but it is more difficult to understand it intuitively. Mathematics involved in this metric processing is beyond this

paper´s scope. Although it is also based on the number of common items between two users, this measure takes into consideration how rare the intersection of the evaluated items is (Owen, et al. 2010).

To illustrate this, let's consider two movie fans that evaluated the movies "Star Wars" and "Casablanca". If they evaluate hundreds of movies, the fact that they evaluated these two is not relevant, because many people have watched them. If both have evaluated a few movies, the fact that they have watched these two movies is considered relevant, because it is not usual that a fan of "Star Wars" is also fan of "Casablanca".

Table 2.4 Similarities using Log-Likelihood related to user 1

	Item 101	Item 102	Item 103	Item 104	Item 105	Item 106	Item 107	Similarities with user 1
User 1	**X**	**X**	**X**					0.90
User 2	**X**	**X**	**X**	**X**				0.84
User 3	**X**			**X**	**X**		**X**	0.55
User 4	**X**		**X**	**X**		**X**		0.16
User 5	**X**	**X**	**X**	**X**	**X**	**X**		0.55

source: (Owen, et al. 2010)

For the purpose of experiments, the Log-Likelihood will be used as the metric that does not take into consideration the evaluations done by the users.

2.4.7. Evaluation of recommender systems

In characterizing recommender systems as a scientific research, it is vital to understand the methodologies for system evaluation. An efficient way to evaluate recommender systems is through the comparison of the generated predictions and the real evaluations made by the user. This is achieved by suppressing a certain evaluation, after this, the recommender system is used to predict this suppressed evaluation, and finally both values are compared.

The attainment of metrics to evaluate the performance of a recommender system before a wide commercial usage is vital to check if the predictions made will be appropriate for the specific purpose. The most used metrics used in the literature for the evaluation of recommender systems will be presented below.

It is important to highlight that, for each *dataset* or business domain, a specific recommender system may be more appropriate than others. It is only possible to define which recommender system is best applied to a domain through experimentations and the analysis of the results.

2.4.7.1.Root Mean Square (RMS) and Mean Absolute Error (MAE)

In statistics, the Root Mean Square (RMS) is one of the several ways to measure the difference between one estimated value and its actual value. RMS measures the mean square to quantify the difference of estimated values.

The RMS value calculated for a set of values $\{x_1, x_2, \ldots, x_n\}$ is the square root of the arithmetic mean of the squares of the values of the set added.

$$x_{rms} = \sqrt{\frac{x_1^2 + x_2^2 + \cdots + x_n^2}{n}}$$

In statistics, Mean Absolute Error (MAE) is another way to quantify the difference between an estimated value and the actual value. As the name suggests, MAE is the Mean Absolute Error.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i|$$

Table 2.5 Difference between MAE x RMS exemplify the differences between RMS and MAE.

Table 2.5 Difference between MAE x RMS

	Item 1	Item 2	Item 3
Current	3.0	5.0	4.0
Estimated	3.5	2.0	5.0
Difference	0.5	3.0	1.0
MAE	= (0.5 + 3.0 + 1.0) / 3 = 1.5		
RMS	$=\sqrt{((0.5^2 + 3.0^2 + 1.0^2) / 3)} = 1.8484$		

2.4.7.2.Precision, Recall, and Fall-Out

Precision measures the precision of the recommendations made by recommender systems and it is measured by the quantity of recommended items that are actually interesting to the user in comparison with the set of all recommended items (Figure 2.6 Precision
). The precision of a system shows how close the prediction is to the actual evaluation done by the user.

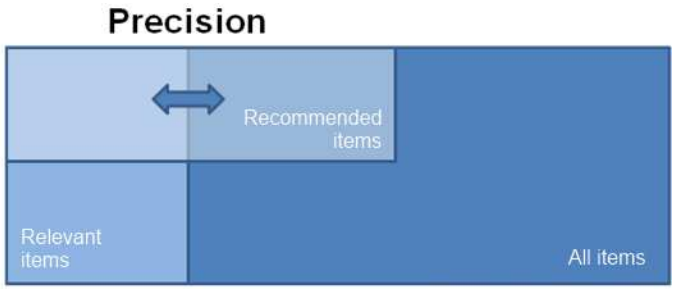


Figure 2.6 Precision

source: (Owen, et al. 2010)

$$precision = \frac{|\{relevant\ items\} \cap \{recommended\ items\}|}{|\{recommended\ items\}|}$$

Recall indicates the quantity of interesting items to the user that appear in the recommendation list in comparison with all relevant items.

Figure 2.7 Recall

$$recall = \frac{|\{relevant\ items\} \cap \{recommended\ items\}|}{|\{relevant\ items\}|}$$

Fall-Out is the proportion of non-relevant items that are recommended in comparison to all non-relevant items.

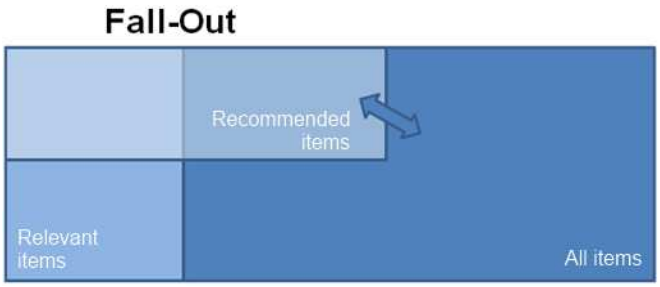


Figure 2.8 Fall-Out

$$fallout = \frac{|\{irrelevant\ items\} \cap \{recommended\ items\}|}{|\{irrelevant\ items\}|}$$

3. SUGGESTED SOLUTION

3.1. Introduction

One of the aims of a recommender system is to generate relevant recommendations. In order to accomplish that, collaborative filtering uses massive datasets, no matter the item types. The recommendations are calculated based on this data (Figure 3.1 Top-Down view of traditional recommender systems
). But working with massive data volume represents a scalability problem that should not be underestimated. This project intends to test a way to decrease user x item matrix space and check if, this way, the recommender system is able to improve the relevance of the recommended items and minimize sparsity issues, super-specialization and lack of surprise (section 2.6.4).

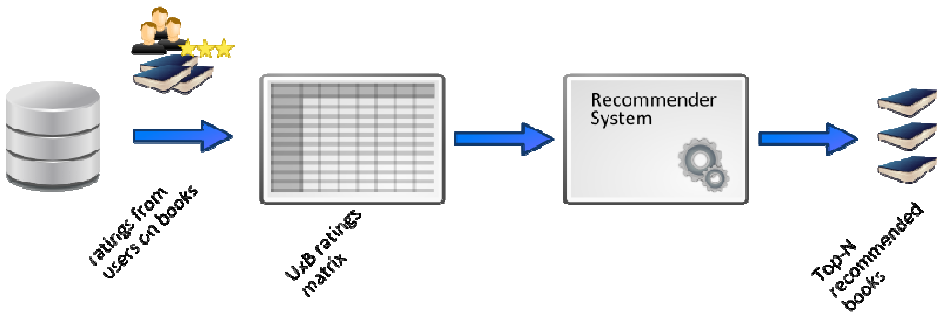


Figure 3.1 Top-Down view of traditional recommender systems

3.2. Separation Degree

As mentioned in the social networks basics, social network members and the relationship established among them form a graph where it is possible to extract their separation degree or, according to graph theory, their distance.

In this paper, the separation degree is taken into account as a natural grouping factor, defined by the members themselves, as people tend to get closer to others who have interests in common (Mendes 2008). Following this reasoning, the smallest is the distance among the network members, and the biggest is the similarity of their interests.

This way, we could represent this closeness among the members in a distance matrix (Figure 3.2 Distance matrix
) and group people who are more similar to each user.

Distance matrix and heat map

	u1	u2	u3	u4	u5	u6	u7	u8	u9	u10
u1	0	1	6	2	1	3	1	2	5	3
u2	1	0	3	1	4	5	2	6	3	4
u3	6	3	0	1	4	2	5	3	2	4
u4	2	1	1	0	3	2	5	3	5	3
u5	1	4	4	3	0	5	4	1	2	4
u6	3	5	2	2	5	0	2	6	3	1
u7	1	2	5	5	4	2	0	2	3	6
u8	2	6	3	3	1	6	2	0	2	1
u9	5	3	2	5	2	3	3	2	0	5
u10	3	4	4	3	4	1	6	1	5	0

Figure 3.2 Distance matrix

These groups are naturally formed by the separation degree and are used for massive data partitioning, thus, decreasing its space.

With the combination of these two ideas, we have a matrix representation of item x user (**U** x **I**), for the **u1**, partitioned this way (Figure 3.3 Top-Down matrix U x I, with evaluations, partitioned taking into account u1 separation degre

Partitioned UxI matrix

	l1	l2	l3	l4	l5	l6	l7	l8	l9	l10
u1	3	5	2	5	4	2	5	4	5	3
u2	4	5	5	2	4	2	2	3	3	1
u5	1	3	4	2	4	4	5	2	3	5
u7	5	4	1	4	5	4	3	1	2	4
u4	2	2	1	3	2	4	4	2	5	4
u8	2	1	4	1	2	3	2	4	2	1
u6	4	5	2	3	1	1	1	2	2	1
u10	5	1	1	2	4	3	4	2	3	5
u9	2	4	2	4	3	3	1	4	5	2
u3	4	5	1	4	4	4	3	1	5	3

1st separation degree
2nd separation degree
3rd separation degree
4th separation degree
5th separation degree
6th separation degree

Figure 3.3 Top-Down matrix U x I, with evaluations, partitioned taking into account u1 separation degre

But the exclusion of items that have been evaluated by more distant users could lead to super-specialization issues or lack of surprise related to the generated

recommendations (section 2.4.4). In an attempt to solve these problems, we have to add another concept used in this solution, that will be explained in the next section.

3.3. Sequential Imputation

Another problem of recommender systems (section 2.4.4) derives from the sparsity of the evaluation in matrix **U** x **I**. Among possible solutions (Soares 2007), the use of missing values imputation may create noise and ends up compromising the recommendations instead of improving them.

But if the imputation is done — in a reduced space — among grouped users that allegedly have high similarity, it may be possible to input values with a reduced noise level.

In order to ease sparsity problems, super-specialization and lack of surprise (section 2.4.4), this solution suggests the use of the underlying idea in partitioning through the separation degree of the social network members, plus the underlying idea in sequential imputation as follows (Figure 3.4 Top-Down view of the recommender system based on separation degree to do sequential imputation
)(Figure 3.5 Process of SocialBased Recommendation
):

1. One of the studied collaborative filtering is chosen to be used for the estimate for user evaluation values of an item. From now on, it will be called the auxiliary collaborative filtering of this system.
2. Every time the system finds a separation degree, it tries to input the user missing evaluations at the same distance to the user to whom the system is recommending. For the imputation, only the evaluations in this space and the auxiliary collaborative filtering are used.
3. The system is thus reefed with new users of the following degree, their evaluations and previous degree imputed evaluations.
4. Steps 2 and 3 are repeated until the desired separation degree.
5. With the resulting matrix, the actual evaluations and the ones imputed by the system, the auxiliary collaborative filtering is used to create the recommendations for the user.

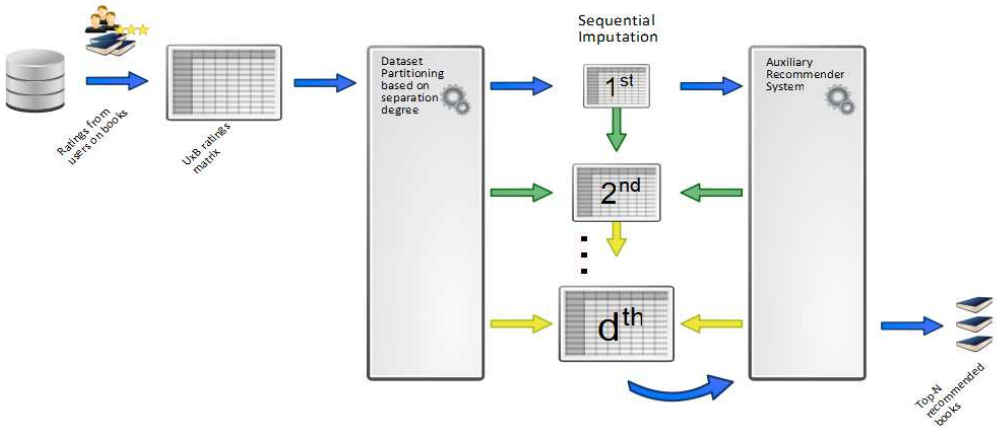


Figure 3.4 Top-Down view of the recommender system based on separation degree to do sequential imputation

With the suggested solution, we move forward to the implementation phase, tests and result analysis.

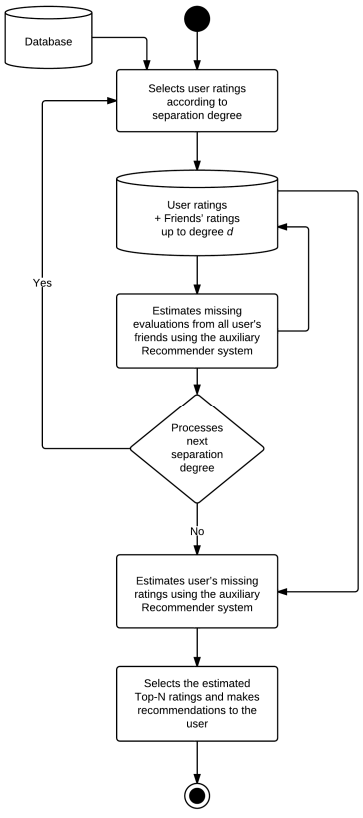


Figure 3.5 Process of SocialBased Recommendation

4. TESTS AND RESULTS

4.1. DATA SUMMARY

From the massive volume of data, 100 users have been randomly selected to form the group of root users of the social network.

Table 4.1 User set

	Total of Users	
Groups	edges	nodes
Root	-	85
1st degree	3,156	3,140
2nd degree	9,427	6,557
Total	12,583	9,782
Total of books:		441,311
Total of evaluations:		2,128,782
Average quantity of books/evaluations per user:		218

To enable experiment processing, the data space had to be reduced. Only the 500 most popular books have been selected. This has defined the data space that will be used for tests as follows:

Table 4.2 Reduced user set

	Total of Filtered Users	
Groups	edges	nodes
Country	-	85
1st degree	2,927	2,911
2nd degree	8,587	5,914
Total	11,514	8,910
Total of books:		500
Total of evaluations:		338,690
Average quantity of books/evaluations per user:		38

4.2. Summary of experiments

4.2.1. Tests Using RMS and MAE

For each recommender system to be tested, the following parameters are alternated as follows:

- The recommender system itself: GenericItemBased, GenericUserBased, KnnItemBased, SlopeOne, SVD, SocialBased;

- Database percentages that will be used for training and test: 60%-40% and 70%-30%;
- The quantity of root users, using the following values: 10, 25, 40, 55, 70, 85 users;
- The quantity of books, using the following values: 50, 100, 150 e 200 books. The used books were the most evaluated ones by root users and their descendents, therefore, the most popular books;
- Three rounds of tests have been made for each combination.

The used values have been chosen because they were the most common ones found in literature, following our advisor suggestions.

4.2.2. Tests USING Precision, Recall AND Fall-Out

These tests try to find a relation between relevant books for the user and the ones recommended by the system.

The calculated measures are precision, recall and *fall-out*, which have been explained in this paper basics (section 0). For these tests, each recommender system has been used to recommend 10 books to each of the root users. The quantity of books and users were varied. The quantity of books and users variation were the same as previous tests.

The list of items, which is considered as relevant to each tested user, is formed by books that have an evaluation value greater than the defined threshold, calculated by the average plus a standard deviation.

The chosen quantity, to be recommended to each tested user, was 10 books. This quantity has been chosen because it is the most common one found in several different sources (newspaper, radio, television, internet etc.). Therefore, Top-10 is the easily understandable measure for the tests that have been run.

It is worth clarifying that the training percentages are not taken into account in this type of test since the evaluation is done as if the system were fully operational. Therefore, all other user evaluations and all tested user evaluations are used, except for the evaluations of books considered by the system as relevant to the user and books that have been separated for system evaluation.

4.2.3. Summary of test result analysis

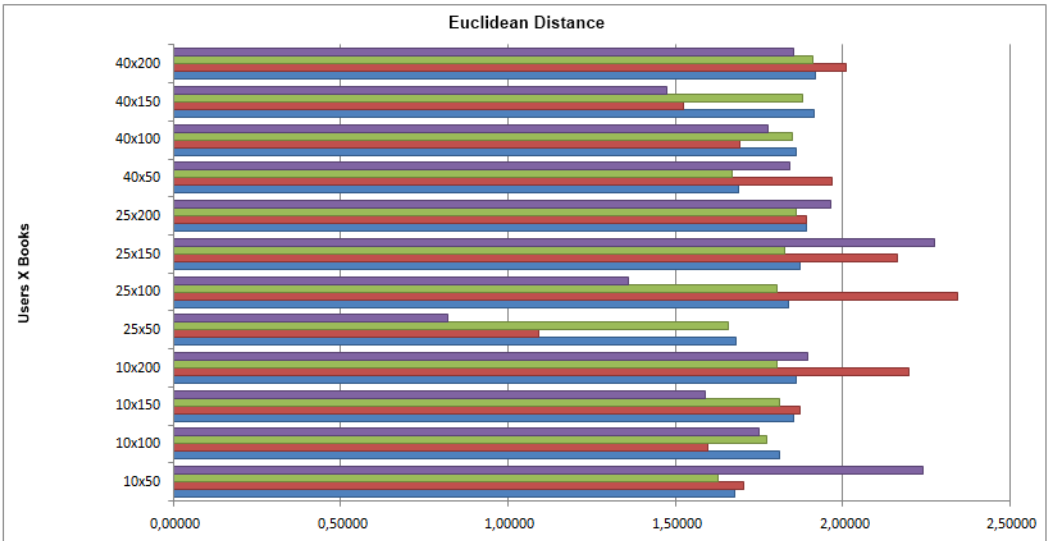
In addition to analyzing all general results of all tested recommender systems, we compared separately UserBased (k-NN) against SocialBased (using UsedBased itself (k-NN) as an auxiliary recommender system), since this is the most frequently used implementation (Ferlin 2008), to check the benefits brought by the suggested solution to traditional implementations.

4.3. Generic User Based (K-NN) x Social Based

With just a few users, the SocialBased recommender system, implemented in this project, suffers a drawback if compared against the basic implementation of

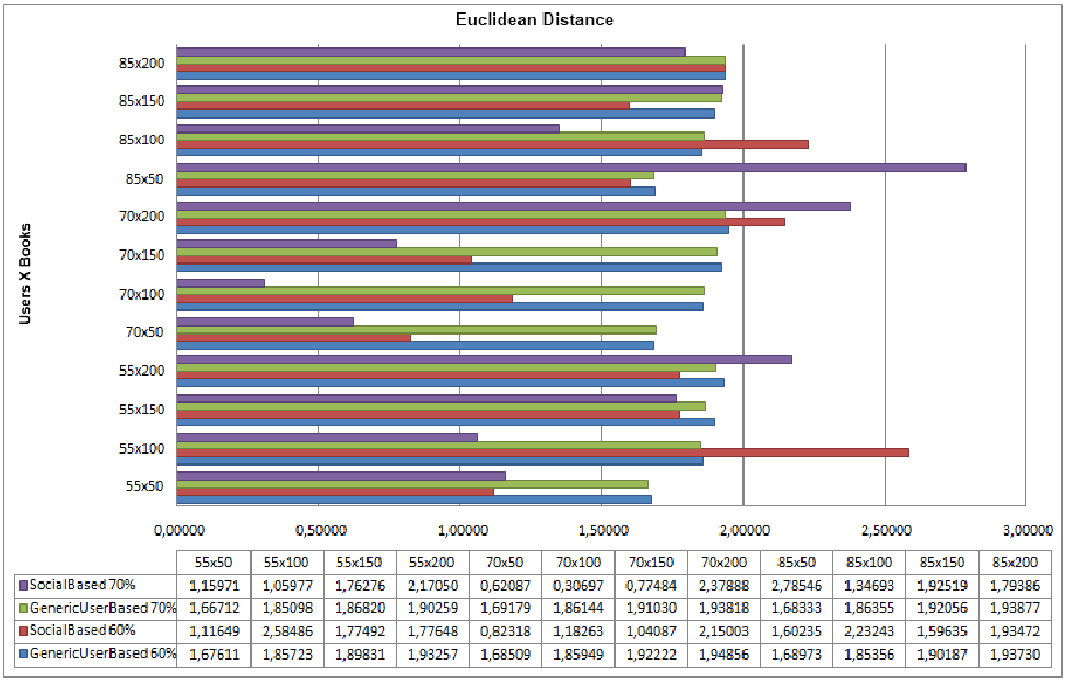
USerBased (K-NN). In all other experiments, SocialBased has been able to enhance the precision of recommendations. The result of the 85x200 combination is contradictory to other results and needs to be checked with a higher number of rounds.

4.3.1. Root Mean Square



	10x50	10x100	10x150	10x200	25x50	25x100	25x150	25x200	40x50	40x100	40x150	40x200
SocialBased 70%	2,24035	1,74892	1,58827	1,89711	0,82129	1,35812	2,27291	1,96351	1,84089	1,77712	1,47622	1,85143
GenericUserBased 70%	1,62600	1,77198	1,81002	1,80477	1,65704	1,80210	1,82649	1,86049	1,67053	1,84893	1,87854	1,90963
SocialBased 60%	1,70232	1,59685	1,87096	2,19619	1,09330	2,34293	2,16164	1,89099	1,96825	1,69321	1,52474	2,01090
GenericUserBased 60%	1,67794	1,81104	1,85353	1,86172	1,68151	1,83649	1,87290	1,89332	1,68915	1,86216	1,91313	1,91855

Figure 4.1 RMS measure with Euclidean distance (a)



	55x50	55x100	55x150	55x200	70x50	70x100	70x150	70x200	85x50	85x100	85x150	85x200
SocialBased70%	1,15971	1,05977	1,76276	2,17050	0,62087	0,30697	0,77484	2,37888	2,78546	1,34693	1,92519	1,79386
GenericUserBased70%	1,66712	1,85098	1,86820	1,90259	1,69179	1,86144	1,91030	1,93818	1,68333	1,86355	1,92056	1,93877
SocialBased60%	1,11649	2,58486	1,77492	1,77648	0,82318	1,18263	1,04087	2,15003	1,60235	2,23243	1,59635	1,93472
GenericUserBased60%	1,67611	1,85723	1,89831	1,93257	1,68509	1,85949	1,92222	1,94856	1,68973	1,85356	1,90187	1,93730

Figure 4.2 RMS measure with Euclidean distance (b)

4.4. Performance evaluation

As we can check in Figure 4.3 Response time for each Recommender system , the implementations done with Slope-One and UserBased (k-NN) have shown to be more efficient than other Recommender systems. However, the response time of other Recommender systems would not preclude its usage in a production environment.

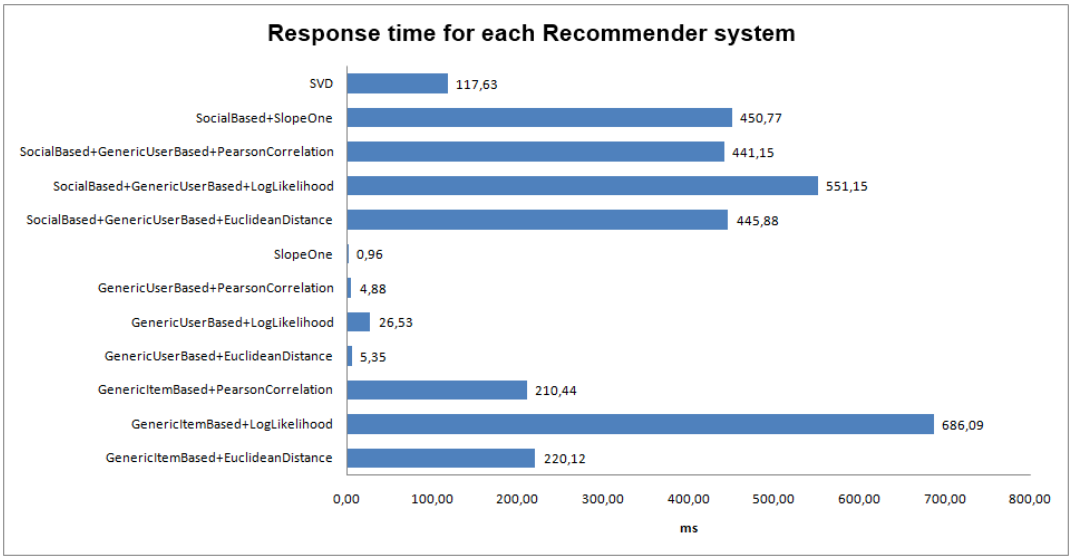
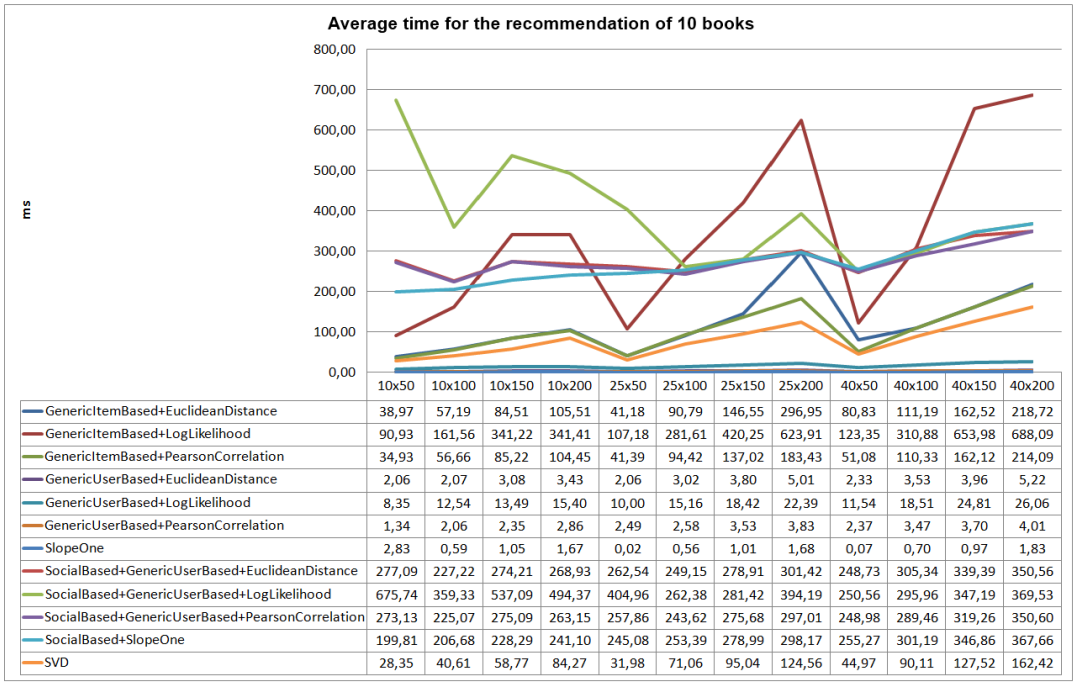


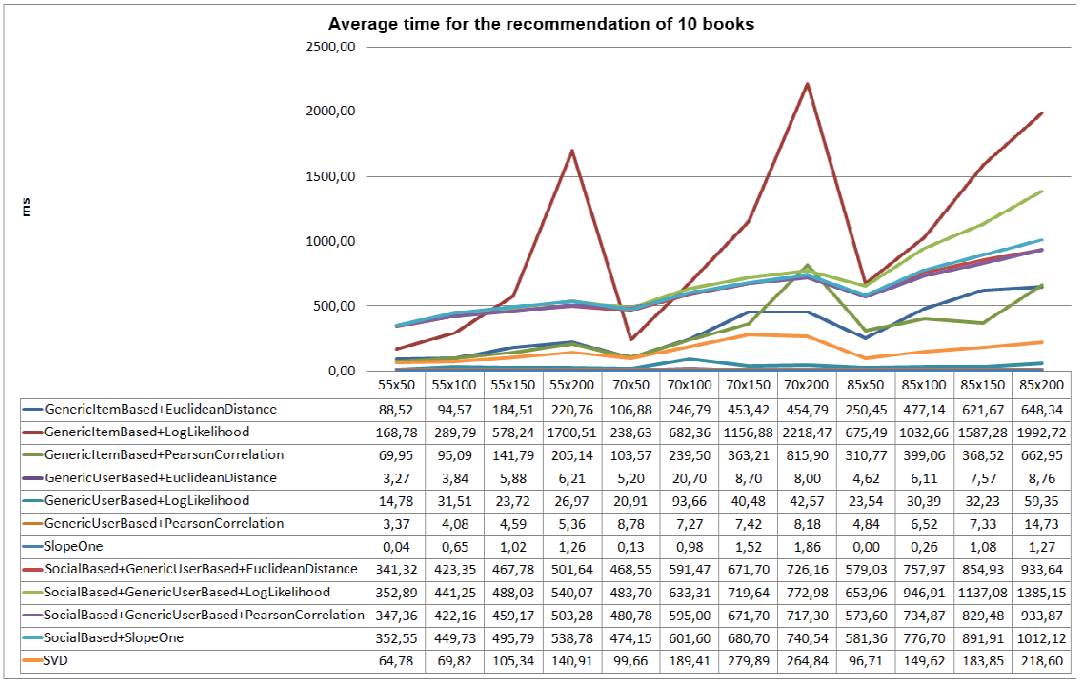
Figure 4.3 Response time for each Recommender system

In Figure 4.4 Average time for the recommendation of 10 books (a) and Figure 4.5 Average time for the recommendation of 10 books (b) are presented the measured average times for each tested user-book combination.



	10x50	10x100	10x150	10x200	25x50	25x100	25x150	25x200	40x50	40x100	40x150	40x200
GenericItemBased+EuclideanDistance	38,97	57,19	84,51	105,51	41,18	90,79	146,55	296,95	80,83	111,19	162,52	218,72
GenericItemBased+LogLikelihood	90,93	161,56	341,22	341,41	107,18	281,61	420,25	623,91	123,35	310,88	653,98	688,09
GenericItemBased+PearsonCorrelation	34,93	56,66	85,22	104,45	41,39	94,42	137,02	183,43	51,08	110,33	162,12	214,09
GenericUserBased+EuclideanDistance	2,06	2,07	3,08	3,43	2,06	3,02	3,80	5,01	2,33	3,53	3,96	5,22
GenericUserBased+LogLikelihood	8,35	12,54	13,49	15,40	10,00	15,16	18,42	22,39	11,54	18,51	24,81	26,06
GenericUserBased+PearsonCorrelation	1,34	2,06	2,35	2,86	2,49	2,58	3,53	3,83	2,37	3,47	3,70	4,01
SlopeOne	2,83	0,59	1,05	1,67	0,02	0,56	1,01	1,68	0,07	0,70	0,97	1,83
SocialBased+GenericUserBased+EuclideanDistance	277,09	227,22	274,21	268,93	262,54	249,15	278,91	301,42	248,73	305,34	339,39	350,56
SocialBased+GenericUserBased+LogLikelihood	675,74	359,33	537,09	494,37	404,96	262,38	281,42	394,19	250,56	295,96	347,19	369,53
SocialBased+GenericUserBased+PearsonCorrelation	273,13	225,07	275,09	263,15	257,86	243,62	275,68	297,01	248,98	289,46	319,26	350,60
SocialBased+SlopeOne	199,81	206,68	228,29	241,10	245,08	253,39	278,99	298,17	255,27	301,19	346,86	367,66
SVD	28,35	40,61	58,77	84,27	31,98	71,06	95,04	124,56	44,97	90,11	127,52	162,42

Figure 4.4 Average time for the recommendation of 10 books (a)



	55x50	55x100	55x150	55x200	70x50	70x100	70x150	70x200	85x50	85x100	85x150	85x200
GenericItemBased+EuclideanDistance	88,52	94,57	184,51	220,76	106,88	246,79	453,42	454,79	250,45	477,14	621,67	648,34
GenericItemBased+LogLikelihood	168,78	289,79	578,21	1700,51	238,63	682,36	1156,88	2218,47	675,19	1032,66	1587,28	1992,72
GenericItemBased+PearsonCorrelation	69,95	95,09	141,79	205,14	103,57	238,50	363,21	815,90	310,77	399,06	368,52	662,95
GenericUserBased+EuclideanDistance	3,27	3,84	5,88	6,21	5,20	20,70	8,70	8,00	4,62	5,11	7,57	8,76
GenericUserBased+LogLikelihood	14,78	31,51	23,72	26,97	20,91	93,66	40,48	42,57	23,54	30,39	32,23	50,35
GenericUserBased+PearsonCorrelation	3,37	4,08	4,59	5,36	8,78	7,27	7,42	8,18	4,84	5,52	7,33	14,73
SlopeOne	0,04	0,65	1,02	1,26	0,13	0,98	1,52	1,86	0,00	0,26	1,08	1,27
SocialBased+GenericUserBased+EuclideanDistance	341,32	423,35	467,78	501,64	468,55	591,47	671,70	726,16	579,03	757,97	854,93	933,64
SocialBased+GenericUserBased+LogLikelihood	352,89	441,25	488,03	540,07	483,70	633,31	719,64	772,08	653,96	946,91	1137,08	1395,15
SocialBased+GenericUserBased+PearsonCorrelation	347,36	422,16	459,17	503,28	480,78	595,00	671,70	717,30	573,60	734,87	829,48	933,87
SocialBased+SlopeOne	352,55	449,73	495,79	538,78	474,15	601,60	680,70	740,54	581,36	776,70	891,91	1012,12
SVD	64,78	69,82	105,34	140,91	99,66	189,41	279,89	264,84	96,71	149,62	183,85	218,60

Figure 4.5 Average time for the recommendation of 10 books (b)

Slope-One and UserBased (k-NN) performance checked in Figure 4.6 Used memory by each Recommender system
has been undermined. Recommender systems with the best performances are the ones which consume more memory. In this item, the SocialBased memory savings were greater than the traditional implementation.

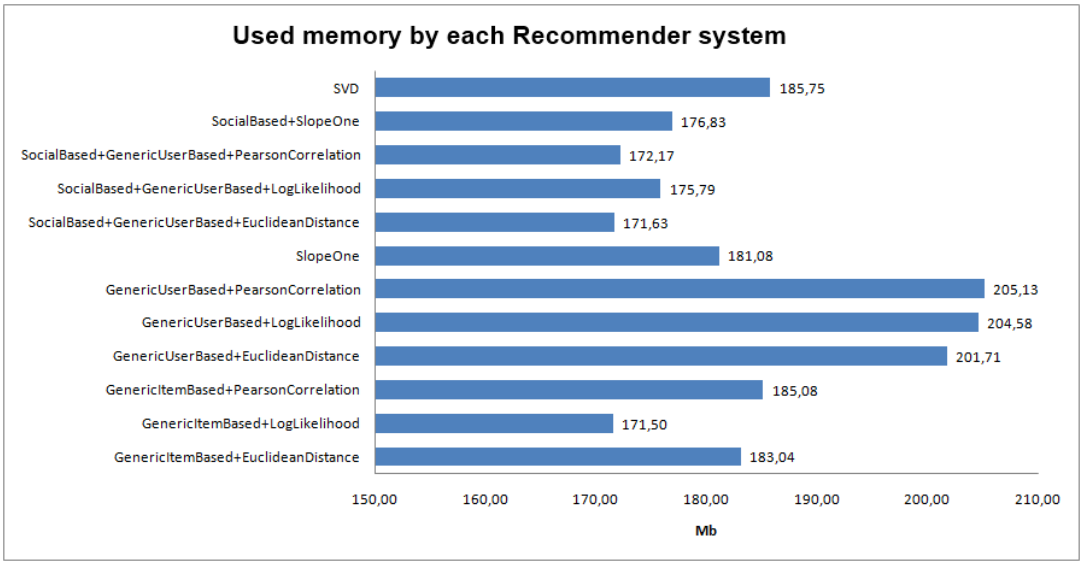
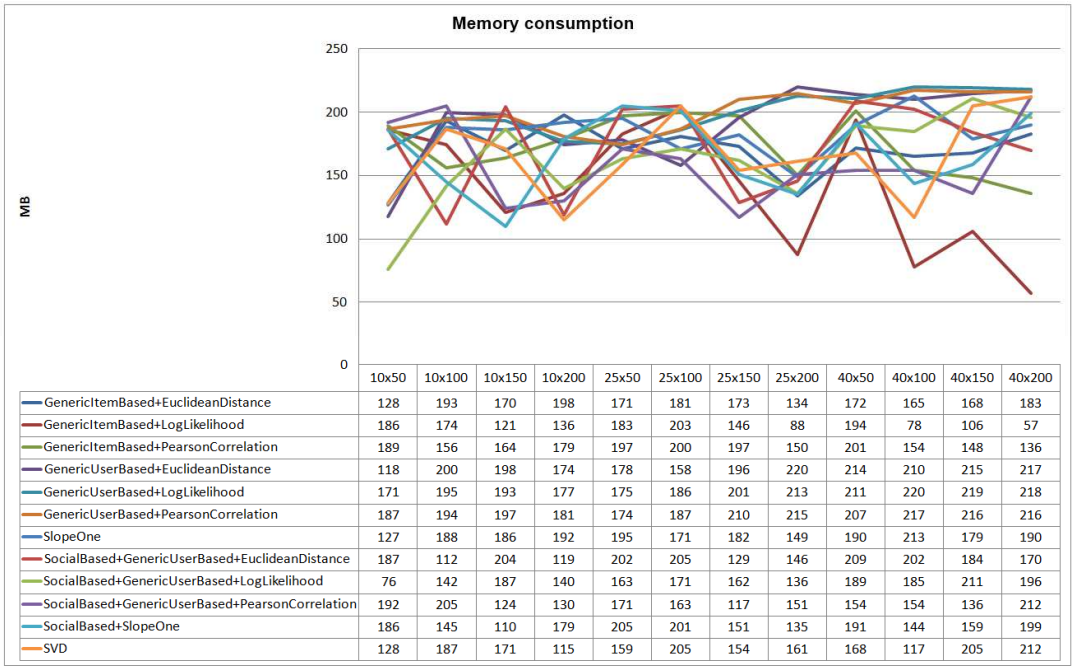


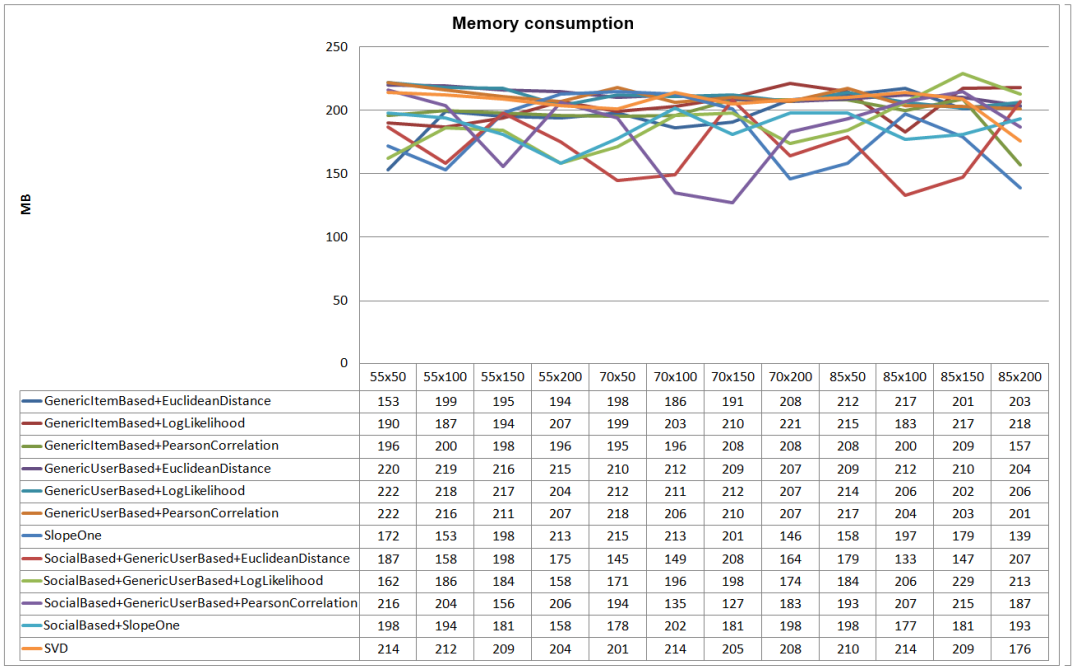
Figure 4.6 Used memory by each Recommender system

In Figure 4.7 Memory consumption (a)
and Figure 4.8 Memory consumption (b)
The average memory consumption can be seen with each tested user-book combination. The average memory consumption of recommender systems was around 185 Mbytes. The values varied between 100 Mbytes and 225 Mbytes.



	10x50	10x100	10x150	10x200	25x50	25x100	25x150	25x200	40x50	40x100	40x150	40x200
GenericItemBased+EuclideanDistance	128	193	170	198	171	181	173	134	172	165	168	183
GenericItemBased+LogLikelihood	186	174	121	136	183	203	146	88	194	78	106	57
GenericItemBased+PearsonCorrelation	189	156	164	179	197	200	197	150	201	154	148	136
GenericUserBased+EuclideanDistance	118	200	198	174	178	158	196	220	214	210	215	217
GenericUserBased+LogLikelihood	171	195	193	177	175	186	201	213	211	220	219	218
GenericUserBased+PearsonCorrelation	187	194	197	181	174	187	210	215	207	217	216	216
SlopeOne	127	188	186	192	195	171	182	149	190	213	179	190
SocialBased+GenericUserBased+EuclideanDistance	187	112	204	119	202	205	129	146	209	202	184	170
SocialBased+GenericUserBased+LogLikelihood	76	142	187	140	163	171	162	136	189	185	211	196
SocialBased+GenericUserBased+PearsonCorrelation	192	205	124	130	171	163	117	151	154	154	136	212
SocialBased+SlopeOne	186	145	110	179	205	201	151	135	191	144	159	199
SVD	128	187	171	115	159	205	154	161	168	117	205	212

Figure 4.7 Memory consumption (a)



	55x50	55x100	55x150	55x200	70x50	70x100	70x150	70x200	85x50	85x100	85x150	85x200
GenericItemBased+EuclideanDistance	153	199	195	194	198	186	191	208	212	217	201	203
GenericItemBased+LogLikelihood	190	187	194	207	199	203	210	221	215	183	217	218
GenericItemBased+PearsonCorrelation	196	200	198	196	195	196	208	208	208	200	209	157
GenericUserBased+EuclideanDistance	220	219	216	215	210	212	209	207	209	212	210	204
GenericUserBased+LogLikelihood	222	218	217	204	212	211	212	207	214	206	202	206
GenericUserBased+PearsonCorrelation	222	216	211	207	218	206	210	207	217	204	203	201
SlopeOne	172	153	198	213	215	213	201	146	158	197	179	139
SocialBased+GenericUserBased+EuclideanDistance	187	158	198	175	145	149	208	164	179	133	147	207
SocialBased+GenericUserBased+LogLikelihood	162	186	184	158	171	196	198	174	184	206	229	213
SocialBased+GenericUserBased+PearsonCorrelation	216	204	156	206	194	135	127	183	193	207	215	187
SocialBased+SlopeOne	198	194	181	158	178	202	181	198	198	177	181	193
SVD	214	212	209	204	201	214	205	208	210	214	209	176

Figure 4.8 Memory consumption (b)

4.5. Discussions about the results

After analyzing the data, we concluded that the bigger the number of users and books is, the bigger the drop in the quality of the recommendations is; this result does

not depend neither on the recommender system nor on the type of evaluation done. This behavior is caused by the increase in data sparsity.

While matrix **U** x **L** (user x book) increases, the results related to Precision and Recall drop. However, the recommended items are not necessarily irrelevant to the user. The evaluation algorithm, used by Precision and Recall, needs to select the relevant items already evaluated by the user (section 4.2.2). This reduces the quantity of relevant items to a small number. If the recommender system suggests items in this list, it means it is working perfectly. However, if it does not recommend items in this list, it does not mean that they are not interesting to the user, but only that they were not among the selected test data used to evaluate the system.

In several combinations of users and books, SocialBased managed to improve the results of traditional implementation, using the sequential imputation and the partitioning of the social network through the separation degree up to the second degree.

Although SocialBased response time has been higher than the majority of the tested system, it does not disable its use. On the other hand, the gains derived from low memory consumption are encouraging, something around 17.5% in comparison with the traditional implementation (k-NN).

It also has been possible to conclude that the variation on the user quantity does not influence the quality of the results as significantly as the increase in the quantity of item does.

The results achieved by RMS and MAE measure the checked mistake between the actual evaluations and the evaluations imputed by recommender systems. However, the goal of a recommender system is not to impute closer values to the actual value, but it is to recommend the more relevant items to a specific user (section 2.4). What matters for the final quality of a recommender system is that the results of Precision, Recall and Fall-out are always the best possible ones.

In run tests, it was found that SocialBased improved UserBased (k-NN) results using the Log-Likelihood similarity metrics. For k-NN-based recommender systems that use this metrics, the use of SocialBased could be beneficial.

5. FINAL CONSIDERATIONS

The focus of this paper was the study of Recommender Systems in social networks. There is a need for tools to help the users handle the great deal of information they receive.

Based on this context, Recommender Systems technologies have been presented and discussed, in addition to the way they are implemented and evaluated and other needed knowledge for its use, such as several similarities metrics and evaluation metrics.

Some studies about the way social networks are formed and their organization also have been discussed. We also presented the small world theory and the idea that people who are closer to an individual have more influence over this individual's opinions. At last, we presented the idea of sequential imputation, the reuse of previously imputed value in a new iteration.

Based on this researched knowledge, we built a "Hot-Deck" solution combining these isolated ideas, in an attempt to add value to current Recommender systems.

Through the implementation of a testing environment and the analysis of the obtained results, we could conclude their advantages and disadvantages. These conclusions are found in section 4.5.

With this paper, we noticed how data related to members of social networks influence Recommender Systems. It is a new different way of partitioning users and defining who the closest people to an individual are in Recommender Systems context.

One of the contributions of this paper is a new *dataset*, which can be used for the continuation of this line of research. Differently from other available *datasets*, this one has information on the relationships among users that can be used to create a graph of a social network.

The natural evolution of this paper would be the expansion of the database so that we could reach a third degree of separation between the users. This way, we could check the behavior of the suggested solution with a higher number of iterations.

There is also the possibility of changing the suggested proposition so that the distribution of processing in several computers can be done, even if they are in a cloud computing environment. This is possible because the used Mahout framework is ready for its future interconnection with Apache Hadoop framework (Apache Software Foundation 2008).

This paper has focused on the use of collaborative filtering. For future papers, we recommend implementations of Recommender Systems based on contents and that can take into account other data, such as age, sex and location of the participants.

REFERENCES

Amaral, Viviane. (2004) *Redes sociais e redes naturais: a dinâmica da vida.* http://www.rits.org.br/redes_reste/rd_tmes_fev2004.cfm (visited in October 17th, 2009).

Apache Software Foundation (2008). *Apache Hadoop!* July, 2008. http://hadoop.apache.org/ (visited in December 12th, 2010).

Balabanovic, M., and Y. Shoham. (2007) "Fab: Content-based, Collaborative." *Communications of the ACM.*

Chisnall, P. (1973) "Marketing Research: Analysis and Measurement." McGraw-Hill.

Comission, U. N. S., and E. C. (2000) For Europe. *Glossary of Terms on Statistical Data Editing.* http://stats.oecd.org/glossary/download.asp (visited in June 17th, 2010).

Cover, T. M., and Hart P. E. (1974) "Nearest Neighbor Classifiers." *IEEE Transactions on Computers.* by Sola Pool, Ithiel, and Manfred Kochen.(1978) ""Contacts and influence." Social Networks."

Farhangfar, A., L. Kurgan, and W. Pedrycz. (2007) "A Novel Framework for Imputation of Missing Values in Databases." *IEEE Transactions on Systems, Man, and Cybernetics*. 692-709.

Ferlin, Claudia. (2008) "Imputação Multivariada: Uma Abordagem em Cascata." Rio de Janeiro, RJ.

Gelman, A., and J. Hill. (2006) "Data Analysis Using Regression and Multi-level / Hierarchical Models." Cambridge University Press, 2006.

*GroupLens Research. (*2010). http://www.grouplens.org/ (visited in June 17th, 2010).

Gurevich, Michael. (1961) "The Social Structure of Acquaintanceship Networks." Cambridge, 1961.

J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. (1999) "An algorithmic framework for performing collaborative filtering." *In Proc. of SIGIR.*

J. S. Breese, D. Heckerman, and C. Kadie. (1998) "Empirical analysis of predictive algorithms for collaborative filtering." In Proc. of UAI.

Kajimoto, Allan Panossian, Renato Shirakashi de Sousa, Sidney Eduardo Serra Zanetti, and Victor Miranda Cirone (2007). "Sistemas de recomendação de notícias na Internet baseados em filtragem colaborativa." São Paulo: IME

Kim, K. Y., B. J. Kim, and G. S. Yi. (2004). Reuse of imputed data in microarray analysis increases imputation efficiency BMC Bioinformatics 2004, 5:160 BMC Bioinformatics

Lakshminarayan, K., S. Harp, and T. Samad. (1999) "Imputation of Missing Data in Industrial Databases." *Applied Intelligence.* 259-275.

Lemire, Daniel, and Anna Maclachlan.(2005) "Slope One Predictors for Online Rating-Based Collaborative Filtering." February 7th

Lepkowski, J., T. Raghunathan, P. Solenberger, and J. Van Hoewyk. (2001) "A Multivariate Technique for Multiply Imputing Missing Values Using a Sequence of Regression Models." Canada, 85-95.

Likert, Rensis. (1932) "A Technique for the Measurement of Attitudes." *Archives of Psychology.*

Lorenzi, Fabiana. (2006) "Sistemas de Recomendação Filtragem Colaborativa e Baseada em Conteúdo." Rio Grande do Sul: UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL - INSTITUTO DE INFORMÁTICA,

Marteleto, M. R. (2001) *Análise de Redes Sociais.* Brasília

Mendes, Regina. (2008) "Ação de Professores em Contexto de Globalização: um estudo a partir do grupo de educação Sócio-ambiental da Pampulha." *Doctorate Thesis in Education - UFMG.* Belo Horizonte, MG, 2008. 67-74.

Milgram, Stanley. (1967) "The Small World Problem." *Psychology Today.*

Montaner, M., B. Lopez, andJ. L. de La Rosa. (2003) "A Taxonomy of Recommender Agents on the Internet." *Artificial Intelligence Review.* June.

Oudshoorn, C., S. Van Buuren, and J. Van Rijckevorsel. (1999) "Flexible multiple imputation by chained equations." *Netherlands Organization for Applied Scienti.*

Owen, Sean, Robin Anil, Ted Dunning, and Ellen Friedman. (2010) *Mahout in action.* Greenwich, CT: Manning.

P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. (1994) "Grouplens: an open architecture for collaborative filtering of netnews." In Proc. of ACM CSCW.

Resnick, Paul, and Hal R. Varian. (1997) "Recommender systems." ACM Communications, March v40 n3 p56(3)

Sarwar, Badrul, George Karypis, Joseph Konstan, and John Riedl. (2001) "ItemBased Collaborative Filtering Recommendation Algorithms." *GroupLens Research Group/Army HPC Research Center.* Minneapolis: Department of Computer Science and Engineering University of Minnesota.

Schafer, J. L. (1997) "Analisys of Incomplete Multivariate Data." Vol. 1. Rio de Janeiro, RJ: Chapman and Hall-CRC.

Soares, Jorge Abreu (2007) "Pré-Processamento em Mineração de Dados:Um Estudo Comparativo em Complementação." Rio de Janeiro, RJ, May/2007.

Vanbuuren, S., J. Brand, C. Groothuis-Oudshoorn, and D Rubin. (2006) "Fully conditional specification in multivariate imputation." *Statistical Computation and Simulation.* 1049-1064.

Verboven, S., K. V. Branden, and P. Goos. (2007) "Sequential imputation for missing values." *Comput. Biol. Chem.* 320-327.

Wang, Jun, Arjen de Vries P., and Marcel Reinders J.T. (2005) "Information and Communication Theory Group." Amsterdam, The Netherlands.

Watts, D. J. (2003) *Six Degrees: The Science of a Connected Age.* 1st. New York: W.W. Norton & Company.

Webber, Rosina Lee. (1998) “Pesquisa Jurisprudencial Inteligente.” *Engineering Doctorate Thesis - Universidade Federal de Santa Catarina.* Florianópolis: UNIVERSIDADE FEDERAL DE SANTA CATARINA, May 14th.

Xue, G. R., C. Lin, Q. Yang, W. Xi, J. Zeng, and Z. Chen. (2005) “Scalable collaborative filtering using cluster-based smoothing.” In Proceedings of the 2005 ACM SIGIR Conference, Salvador, Brazil.