UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

GUILHERME MEDEIROS MACHADO

AwARE: an Approach for Adaptive Recommendation of rEsources

Thesis presented as a partial requirement to obtain the degree of Doctor in Computer Science.

Advisor: Prof. Dr. José Palazzo Moreira de Oliveira

Porto Alegre
2018

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Medeiros Machado, Guilherme
Normas para Apresentação de Teses do Instituto de Informática e do PPGC [manuscrito] / Guilherme Medeiros Machado. – 2018.
100 f.:il.
Orientador: José Palazzo Moreira de Oliveira;
Tese (Doutorado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR – RS, 2018.
1.ABNT. 2.Processadores de Texto 3.Formatação eletrônica de documentos. I. de Oliveira, José Palazzo Moreira. II. AwARE: an Approach for Adaptive Recommendation of rEsources.

"...Le monde n'est plus le même, personne n'y peut rien; il faut essayer de s'adapter."

—Simone de Beauvoir
Les Mandarins.1954

ACKNOWLEDGMENTS

Gostaria de agradecer ao meu orientador José Palazzo, por todos os conselhos, orientação, incontáveis reuniões, pela seriedade e paciência mantidos durante os dois anos de mestrado e agora os quatro anos de doutorado, um exemplo que pretendo seguir em minha carreira acadêmica.

Gostaria de agradecer a Lucy, a Leticia, e ao Evaldo, por estarem sempre presentes, pelos momentos compartilhados na vida, pelo amor, carinho, pela dedicação de vocês, certamente esse trabalho não seria possível sem vocês.

Gostaria de agradecer também aos amigos que me acompanharam durante o sanduíche: Vinícius Wohnrath, Itzi Gaëlle, e Adly Manseri. Vocês se tornaram uma família para mim, sem vocês certamente a experiência em Paris não seria tão enriquecedora, e esse trabalho seria mais difícil de ser finalizado.

Un merci aussi à Amel Bouzeghoub, ma professeure pendant le période du sandwich. Merci pour tous vos conseils, pour la patience, et le super accueil en France.

Um muito obrigado a todos os amigos que estiveram sempre de ouvidos e corações abertos nesses anos, esse trabalho se tornou mais fácil por causa de vocês: Vinícius Wolozsyn, Paloma Bampi, Diana Corso, e Júlia Kikuye.

Um muito obrigado também aos amigos de laboratório Vinícius Maran, Isabela Gasparini, Oscar Ortegon, Carlos Habekost, Gabriel Lunardi e todos os outros que passaram pelo laboratório 213 nesse período.

Obrigado ao Instituto de Informática da UFRGS. E um muito obrigado ao CNPq e à CAPES por financiarem parcialmente essa pesquisa.

RESUMO

Sistemas de recomendação foram propostos no início da década de 1990 com o objetivo de auxiliar seus usuários a lidar com a sobrecarga cognitiva criada com o advento da internet e o aumento constante de documentos. De lá para cá tais sistemas passaram a assumir vários outros papéis, tais como "auxiliar usuários a explorar", "melhorar a tomada de decisão", ou até mesmo "entreter". Para atingir tais novos objetivos, o sistema necessita olhar para características do usuário que auxiliem no entendimento da tarefa desempenhada pelo usuário e como a recomendação pode auxiliar tal tarefa. Nesse sentido, propõe-se nessa tese uma integração entre estratégias de recomendação e de adaptação para criar um novo processo de recomendação adaptativa. É mostrado que tal integração pode melhorar a acurácia da recomendação, e dar bons resultados na retenção de usuários, e na interação destes com os sistemas. Para validar a abordagem, é implementado um protótipo para recomendação de filmes a serem utilizados em sala de aula. São também coletadas estatísticas de 78 usuários que participaram do experimento de avaliação da abordagem.

Palavras-chave: Sistemas de Recomendação. Sistemas Adaptativos. Recursos Educacionais. Acurácia.

AwARE: an Approach for Adaptive Recommendation of rEsources

ABSTRACT

Recommender systems were proposed in early 90's with the goal to help users deal with cognitive overload brought by the internet and the constant increase of documents. From there to now such systems have assumed many other roles like "help users to explore", "improve decision making", or even "entertain". To accomplish such new goals, the system needs to look to user characteristics that help in understand what the user task is and how to adapt the recommendation to support such task. In this direction, it is proposed in this thesis an integration between recommender and adaptive strategies into a new process of adaptive recommendation. It is shown that such integration can improve recommendation accuracy and give good results to user retention, and interaction with the systems. To validate the approach, it is implemented a prototype to recommend movies to be used in a classroom. It is also collected some statistics about the 78 users who have participated of the experiment for evaluation of the new approach.

Keywords: Recommender Systems. Adaptive Systems. Learning Resources. Accuracy.

LIST OF FIGURES

Figure 2.1 – The Rating Matrix Split Strategy 26
Figure 2.2 – Approaches for context-aware recommendation 29
Figure 2.3 – Taxonomy of adaptation techniques 37
Figure 3.0.1 – The keywords distribution 41
Figure 3.2 – The cluster composition 42
Figure 3.3 – Keywords distribution in each cluster 43
Figure 3.4 – Most common techniques 50
Figure 3.5 – The validation techniques 51
Figure 3.6 – The application domains distribution 53
Figure 4.1 – The strategies for adaptive recommendation 58
Figure 4.2 - Algorithm for adaptive recommendation 62
Figure 4.3 - The rating prediction function 63
Figure 4.4 - The algorithm for similar movies recommendation 63
Figure 4.5 - User tasks in TEL recommenders 65
Figure 4.6 - An instance of the post-adaptation process 67
Figure 5.1 – TWM start page 69
Figure 5.2 – Lesson plan for “12 angry men” movie 70
Figure 5.3 – The prototype database schema 72
Figure 5.4 – The learner profile screen 73
Figure 5.5 – The screen for get user preferences 74
Figure 5.6 – The top-k recommendation screen 75
Figure 5.7 – The movie details screen 76
Figure 5.8 - Accesses by country 79
Figure 5.9 - The system accesses overview 80
Figure 5.10 - Items presentation Satisfaction 81
Figure 5.11 - Recommendations Satisfaction 82
Figure 5.12 - Satisfaction of similar movies 83
Figure 5.13 - User overall satisfaction 84
Figure 5.14 - Learning curve for accuracy in rating prediction 85

LIST OF TABLES

Table 3.1 – Search engines where the string was submitted 40
Table 3.2 – Taxonomy for papers classification 45
Table 3.3 – Papers classification 46
Table 5.1 – The adaptive recommendation purposes 76
Table 5.2 – The metrics related to the recommendation purposes 77
Table 5.3 – Dataset Statistics 78

LIST OF ABBREVIATIONS AND ACRONYMS

AHS	Adaptive Hypermedia Systems
MF	Matrix-Factorization
RS	Recommender Systems
SGD	Stochastic Gradient Descendent
TEL	Technology Enhanced Learning
TF-IDF	Term Frequency-Inverse Term Frequency
TWM	Teach with Movies

SUMMARY

1 INTRODUCTION....................11

2 CONCEPTUAL FOUNDATION....................15

2.1 Recommender Systems....................15
2.1.1 Content-Based Filtering....................16
2.1.2 Collaborative Filtering....................20
2.1.3 Context-aware Recommendation....................27

2.2 Adaptive Hypermedia Systems (AHS)....................29
2.2.1 Adapt to what?....................30
2.2.2 How to provide adaptation?....................34

3 A LITERATURE MAPPING ON ADAPTIVE RECOMMENDATION....................38

3.1 Discussion and Results....................40
3.1.1 Clustering by keywords....................40
3.1.2 Research topic 1: existence of a hybrid approach....................43
3.1.3 Research topic 2: most common techniques....................50
3.1.4 Research topic 3: validation techniques....................51
3.1.5 Research topic 4: application domains....................52

3.2 Conclusions of the Mapping....................54

4 AN APPROACH FOR ADAPTATION IN RECOMMENDER SYSTEMS....................56

4.1 The Adaptive Recommendation Process Overview....................56
4.1.1 Pre and Post Adaptation....................57
4.1.2 Adaptation During Recommendation....................59
4.1.3 Selecting an Adaptation Strategy....................61
4.1.4 Algorithms to adaptive recommendation....................61

4.2 Approach for Adaptive Recommendation of Educational Resources....................64

5 EXPERIMENTS AND RESULTS....................68

5.1 Dataset....................68

5.2 Prototype....................71

5.3 Experiment Results....................76
5.3.1 Data Collect and Statistics....................77

6 CONCLUSIONS AND FUTURE WORKS....................86

6.1 List of Publications....................87

APPENDIX <SATISFACTION SURVEY>....................100

1 INTRODUCTION

Recommender systems were first proposed as a solution to deal with the problem of user cognitive overload, where the amount of information to be analyzed exceeds the user capability. Tapestry (GOLDBERG et al., 1992), one of the first recommender systems, was developed as a platform where its users could collaborate by annotating their reaction on emails they read. The system then checked for similar users to filter their messages in a similar way. Such systems have evolved and started to be used to support the users not only in making annotations in context but for a variety of other tasks, like: (i) “Help users to find objects that match their long-term preferences”, (ii) “Actively notify consumers of relevant content”, (iii) “Show alternatives when the desired item is not available”, (iv) “Show accessories to be used with the desired item”, (v) “Help users explore or understand the item space”, (vi) “Remind users of already known items”, (vii) “Improve decision making, e.g., in terms reduced decision making or higher choice satisfaction”, (viii) “Establish group consensus balancing the interests of different group members”, (ix) “Help users to explore, (e.g., providing a convenient way to browse a catalog)”, and (x) “Entertain the user, providing a satisfying emotional experience when visiting the site”, as well as, some others as shown in (HERLOCKER et al., 2004) (JANNACH; ADOMAVICIUS, 2016).

Despite such versatility of the recommender systems utilization, most of the recent research in the field has been conducted with the goal of mainly enhancing the accuracy in the tasks of rating prediction and item ranking (JANNACH; ADOMAVICIUS, 2016). Such goals are important but do not support the evaluation of aspects related to uses like “Entertainment” or “Help users to explore”. For instance, an enhance in the system accuracy does not guarantee that it provides a convenient way to browse the catalog when there is no shopping intent. Tasks that are not directly related to the accuracy of item rating or ranking (also referred to “traditional tasks”) present a different challenge in the design and evaluation of approaches that support these tasks. How to evaluate, for instance, the effectiveness of a recommender system in improving the user choice satisfaction? One of the characteristics presented by some of such tasks is the need to focus and evaluate different human aspects, the information of user satisfaction (user rating) on a recommended item is not sufficient to measure the user satisfaction on characteristics such as, how the items are presented, how the system provides the navigation between the item catalog, or how the content of an item is shown.

Recently published research has shown a concern of the scientific community towards the intention of leveraging recommendation aspects not related to the traditional tasks (KAPOOR et al., 2015) (HARPER et al., 2015) (EKSTRAND et al., 2015) (PUTHIYA PARAMBATH; USUNIER; GRANDVALET, 2016) (TEO et al., 2016). The main conference of recommender systems, has demonstrated a growing interest for papers not related with accuracy methods is demonstrated by specific tracks such as “Beyond Accuracy” (“RecSys ’16: Proceedings of the 10th ACM Conference on Recommender Systems”, 2016).

A projection for the future of the research in human factors of recommender systems is depicted in (CALERO VALDEZ; ZIEFLE; VERBERT, 2016). The authors have conducted a bibliographic review and their conclusions pointed that one of the aspects that will play an important role in the future recommender applications is the capability of system adaptation. One of the ways to look at the adaptation is considering it as a feature of the system that has the intent of delivering some content in different ways to different users since the users are distinct from each other and should not receive the content in the same way another distinct user would receive. Such problem was named in the domain of Adaptive Hypermedia as “one size fits all” (BRUSILOVSKY, 2001). The problem is broadly discussed in the papers of adaptive hypermedia and the strategies mentioned in these papers could be used as a starting point to address the challenge of providing adaptation to recommender systems.

The adaptation also plays an important role in ubiquitous systems, where it is necessary to deal with a multitude of computer resources and adapt the way of interaction with each different user. Since ubiquitous systems rely on an infrastructure of computing distributed in an environment, it is necessary to consider the user current context to adapt the way of interaction. For instance, if the system needs to communicate with a user inside an intelligent house, it must consider at least the room where the user is and the available communication devices in this room. It is not difficult to find in the literature some approaches that consider context information to adapt the way of communication with the user (KAMBARA et al., 2014) (MACHADO et al., 2013) (OTEBOLAKU; ANDRADE, 2015) (MAEKAWA et al., 2012) (YAO et al., 2016). Besides using the context to support adaptation, recent ubiquitous approaches are applying recommender systems strategies to overcome their challenges. Many examples of approaches that show the integration of recommendation techniques in ubiquitous environments have been found in recent literature (FAN et al., 2015) (SILVA et al., 2012) (BAGCI; KARAGOZ, 2015) (WANG; WU, 2011) (SALMAN et al., 2015) (MACHADO; DE

OLIVEIRA, 2014). Some of these papers call attention because they present ubiquitous approaches that merge characteristics of adaptation and recommendation, they even adapt the way the recommendation respond face an undesirable situation (MACHADO et al., 2013) or adapt the way a recommended message is displayed to the user (KAMBARA et al., 2014). Such approaches present recommendation solutions that are able to treat complex ubiquitous problems as well as provide insights to make recommender systems more adaptable.

To illustrate the advantages of utilizing an adaptive recommender approach, here is a scenario. A student is waiting for her train to go to the university. She wants to receive content about a test she is going to take in a couple of hours. A context-aware recommender system would propose any educational content that best fits the user preferences and is suitable to the user current context (in the train station). For instance, the system will propose some videos, books, and audio about the desired content prioritizing the text material because the user current context is in a public place (the train station) without headphones. Since the user has only her smartphone available to read the recommended material, the system would be expected to provide at least some adaptation in the presentation to consider the small dimensions of the screen. But it would also be interesting if the system manages the item the user has selected to explore; or if it takes in consideration other user profile features, for instance, the user learning goals when selecting the items. The system could, for instance, pre-select and give preference to the presentation of subjects the student has more difficulty in assimilating or could select a list of items already adapted to this need of content. If selecting a book, the system could give preference to those which emphasize the user needs, despite such book not being the user preferred one, or the systems could highlight the most appropriate chapters or sections of another book. Such interface adaptation, profile-aware filtering, and content selection executed by the system over the selected item is an adaptation feature. Such adaptation is, a managing of a specific item or set of items the user will explore, or even the way the items are displayed to fulfill some user need. To perform such adaptations, the system must know the user profile and take in consideration her characteristics when recommending.

To investigate the existence of recommendation approaches that also include such adaptive characteristics in the ubiquitous domain, a mapping of the recent literature has been conducted and is presented in Chapter 3. The goal of this mapping is to find adaptive recommender approaches and learn how adaptation is performed over recommendation in ubiquitous systems. The main conclusions showed the almost inexistence of such approaches and the need for adaptation is intrinsically related with the recommender systems, applied to ubiquitous environments or not.

These conclusions motivated the development of a novel approach and an algorithm for integration of adaptive strategies inside the recommendation loop. These adaptive characteristics are information about the user profile, besides the preferences, that are inserted in the estimation of the recommendation. Such approach is structured and presented on Chapter 4, with the approach is also presented the algorithm to insert the adaptive characteristics in a function of rating prediction. An experiment was performed using a prototype that implements the proposed approach. The results have shown the approach can predict ratings with a considerably gain in accuracy and it was also capable of achieve other goals non-related to "traditional tasks". So, briefly this thesis main contributions are:

- A systematic mapping of the literature on the subject of adaptive recommender approaches to ubiquitous environments.
- An approach for inserting adaptation in the recommendation process. Showing the advantages and drawbacks of each strategy presented.
- An algorithm for including profile features inside the rating prediction.
- A validation using a prototype that implements the approach.
- The collected dataset of user profile and ratings.

The rest of the text is structured as follows, Chapter 2 presents a conceptual foundation of both recommender and adaptive systems domain. Chapter 3, presents the systematic mapping on adaptive recommendation and its conclusions. In Chapter 4, it is structured the approach for insert adaptation in the recommendation process and also is discussed how to apply such definitions on educational recommenders. Chapter 5, presents the implemented prototype and the experiments results. Chapter 6, presents the conclusions and future works.

2 CONCEPTUAL FOUNDATION

This chapter presents and describes the areas of recommender systems and adaptive systems. It is presented the main techniques used in each area to provide personalization.

During the description of Adaptive Hypermedia Systems, it is presented that one of the ways to provide adaptation is by using Information Retrieval techniques to adapt the hypermedia. The authors give only a glimpse of how the use of a recommendation algorithm could help adapt the navigation path of a hypermedia.

Since the ambition of this work is to provide a clearer vision of how the techniques of adaptation can influence the recommendation, we first provide a broader view of each area in the next subsections.

2.1 Recommender Systems

Recommender systems have their roots in the beginning of the 1990's. Such systems were firstly proposed to deal with the problem of cognitive overload many users were experiencing when managing the ever-increasing amount of information made available through personal computers. The first proposition found in the literature is Tapestry (GOLDBERG et al., 1992), which was a system proposed to manage incoming e-mails of a company.

Each employer of the company read the e-mails and then endorse those they found to be relevant for them. The system collects such information and when another user is searching for a specific subject in her inbox, she can verify those messages that have received a bigger number of endorsements from other employees and prioritize such messages. Such collaborative endorsement of messages was named as collaborative filtering and later this become the most popular strategy of recommendation.

There are two main manners of providing recommendation, either by using the intrinsic characteristics of an item to match it with the user preferences, this is known as *content-based filtering*; or by matching users by their items consumption and recommending new items the similar users have chosen and liked, such strategy is known as *collaborative filtering* (JANNACH et al., 2010).

Another important recommender strategy, later proposed by (ADOMAVICIUS et al., 2005), is known as context-aware recommendation. The consideration of the contextual information during the recommendation process was of great importance in some domains where the changing of the context can also change the user preference. For example, the information of location, day and company can change the choice of a movie in a movie recommender system. Consider a man at home, with his girlfriend, on a weekend choosing a movie to watch. Now consider the same man, at home, alone, on a week day, his preferences may change in each situation, he can watch a documentary during the week days but would prefer to watch a comedy during the weekend, for instance. Other example is a person planning a winter vacation, aiming to go to a sunny place, so the weather condition in the destiny would be an important context information to be taken during the estimation of the recommendation.

These three paradigms of recommendation, *content-based*; *collaborative filtering*; and *context-aware*, are better explained in the following subsections.

2.1.1 Content-Based Filtering

Content-based recommendation is an intuitive way of recommending items, that relies basically on a description of the item (known as content), and a user profile containing the user past interactions with the items. One of the drawbacks of such technique is the need of having items well described. For instance, to recommend a movie using this technique it is necessary to provide description like, the genre, the list of actors, the director, a brief description of the movie, between other technical features and characteristics of such item. Other challenge to implementation of content-based recommendation is the discovery of qualitative features, that refers to the reasons someone has liked an item. The qualitative features present an even bigger challenge than the descriptive content, that commonly are provided by manufacturers about the items. Qualitative features in preference domains reflects the reasons why a user has liked an item; this reason sometimes is not related to intrinsic features or characteristics but could be instead related to an exterior design of a product.

Despite such drawbacks, the content-based recommendation presents some advantages facing other recommendation strategies. The first one, is the need to rely only on the item content and a user profile that reflects interactions with the items. This makes the content-based strategy able to provide suggestions of items even if the whole community of users comes down to a single user. Content-based is also able to provide recommendations even with little interaction of the user with the items. So, it is not affected by the classic *cold-start* problem that affect the collaborative strategies. Besides, such strategy is still most fitted to recommendation in domains where the items are texts, or news, or have their features presented in texts. Such recommendable items will be referred as documents.

This ability to treat texts comes from some techniques and algorithms inherited from the field of Information Retrieval, that has played an important role in providing knowledge also to the learning of the user profile in content-based recommendation.

2.1.1.1 TF-IDF and the vector space model

One of the most popular strategies to derive item features is the vector space model and TF-IDF. Consider a document set that has its content represented by vectors containing all the words presented in the set and each time a word is found in a document the position is set with 1, otherwise with 0. In this simple model, the requirement to do a recommendation is to build a user vector with the words the user has interest in set with 1 and match the user vector with the document vector. This naïve approach, however, does not take in consideration the case the longer is the document, the higher is the probability to be recommended. So, it is necessary some strategy to avoid longer documents to be more recommended.

TF-IDF (*term frequency-inverse document frequency*) comes as a proposition to overcome this problem. Instead of describing the documents using Boolean keywords, they are encoded as TF-IDF vectors in a multidimensional Euclidean space. Each keyword now is obtained by the product of two sub-measures; term frequency, and inverse document frequency.

Term frequency (TF) is the number of times a term ‘i’ appears in a document ‘j’, this has the assumption the more frequent a term is more important it is. However, to prevent longer documents from getting a higher relevance is necessary normalize this measure. So, it is calculated as:

$$TF(i,j) = \frac{freq(i,j)}{maxOthers(i,j)} \tag{1}$$

Where freq(i,j) is the number of times the term ‘i’ appears in the document ‘j’ and maxOthers(i,j) represent the number of the maximum frequency between all the other terms present in the document.

Inverse document frequency (IDF) is a measure that aims to reduce the importance of a term that appears frequently in the set of documents and consequently is not discriminative of the target document. IDF is calculated as:

$$IDF(i) = log\frac{N}{n(i)} \tag{2}$$

Where N is the number of all recommendable documents and $n(i)$ is the number of documents of N which i is present. The product of TF and IDF gives the TF-IDF metric used to describe the documents using vectors.

$$TF\text{-}IDF(i,j) = TF(i,j) * IDF(i) \tag{3}$$

After having the items’ profile vector computed we can apply a cosine similarity to get a rank of other similar items. A user profile can also be built using the TF-IDF vector space, this allows the verification of similarities between user and items. Following such strategy the item selection problem in content-based filtering can be described as “recommend items that are similar to those the user liked in the past”(JANNACH et al., 2010).

2.1.1.2 The user profile learning

The next task is to learn a user profile that will be used to classify the interesting items to be recommended to the user. Machine learning algorithms have been successfully used to accomplish such task (DE GEMMIS et al., 2015). The algorithms try to classify the items into pre-defined classes, such as user-likes or user-dislikes, as well as into more complexes classes division schemas. The most common techniques to perform such classification are probabilistic methods, relevance feedback and k-nearest neighbors.

Naïve Bayes is one common probabilistic strategy to classification of documents. The method stores observed data and then generates a probability of a document 'd' belongs to a class 'c'.

Other method to learn the user profile is through Relevance Feedback which stores documents in vectors of TF-IDF. The learning is achieved by combining document vectors (of positive and negative examples) into a prototype vector for each class. After the building of the prototype vectors it is computed a similarity measure between each document and prototype vector. The documents more similar to the prototype are then assimilated into that class.

The Nearest Neighbors are also known as the lazy learner method, because it only stores the vectors the user has interacted with (training data), in memory. Then a similarity function is applied between all stored document and the new unseen document. The "k-nearest neighbors" are selected and returned to the user as a recommendation. The drawback of this method is the inefficiency in time because it does not have a training phase and so the classification must be calculated each time.

2.1.1.3 Advantages and limitations of content-based techniques

Some of the advantages of using content-based filtering are:

- User Independency: the content-based filtering only needs ratings from the own user to recommend items. Differently from the collaborative filtering algorithms that needs to check other user ratings to find the nearest neighbors of the active user. Then the items the nearest users have liked are recommended;
- Transparency: the recommendations can be explained by listing the content features or descriptions of the item that help to understand the reasons to recommend the items. In collaborative filtering, however, the only explanation is that other unknown users also have liked the recommended item.
- New Item: they are able to recommend new items that have not been rated by any user yet. Because of such characteristic, they do not suffer from the first-rater problem, where a new item needs to receive ratings from a substantial number of

user ratings to be recommendable. This happens in collaborative approaches that rely mainly in other users' preferences.

Some known shortcomings are:

- Shallow content analysis: there is no way to guarantee an item is well described with all necessary features to characterize it and mock the user behavior in analyzing and choosing it as a particularly interesting item. This problem is common when extracting features of textual descriptions of items, how to check the quality of an item relying solely in keywords? How to differentiate a well written document from a bad one if they use the same set of keywords? The aesthetic side is also overlooked when the recommendation is based only in textual features. One strategy is to use ontologies to describe the items, but it demands a high effort and even though it is not assured the quality of the description. Another challenge is to find enough information to discriminate the item. Some problems of textual feature extraction are related to: Polysemy (one word with multiple meanings); Synonymy (multiple words with same meaning); Multi-word expressions (when a combination of words have a different meaning from the isolated ones); Entity identification (the difficulty to identify names of persons, organizations, locations, etc., in the text); Entity linking (the difficult of identify entity references in the text);
- Over-specialization: the nature of recommend only items similar to the ones the user is familiar with, or lack of serendipity, is a known drawback in content-based recommenders. Serendipity is also known as the capacity of the system recommend good unexpected items the user probably would not get to know without the recommendation. Content-based algorithms are known for offer more of the same.
- New user: despite being known for not suffering from cold start, in content-based it is necessary an enough number of ratings from the user before the system being able to understand the preferences and provide accurate recommendations.

2.1.2 Collaborative Filtering

Collaborative filtering approaches rely on user past behavior or the opinions of an existing user community to provide recommendations to the current user. These type of recommender systems are widespread in many industry applications such as online retail sites, media streaming services and educational platforms. The collaborative approach is well known and form many years its algorithms and techniques have been studied. The pure collaborative approach takes a matrix of user x item ratings as input and produce two outputs: (i) a prediction of ratings the user would give to a certain item; and (ii) a ranking of the top-k most likeable items. The ranking traditionally does not contain items the user has selected previously. Traditionally, because recent research efforts have proposed a mixt ranking of familiar and new items in the ranking to promote user-system fidelity (KAPOOR et al., 2015).

The collaborative approaches can be divided in neighborhood-based and model-based approaches. There are two possibilities to recommend utilizing a neighborhood-based strategy. The first is to take the similar users ratings as basis to estimate the current user rating to an item; this method is called user-based. The second is to consider the ratings the current user gave to similar items, to estimate the current item rating, two items are considered similar if their ratings between the users of the system are similar, and this method is called item-based.

Model-based approaches, have their recommendations generated by looking to a portion (training set) of the user' ratings and learning a behavior model. Such model is then used to retrieve the user ratings for new items. One characteristic of such approaches is they are more accurate than neighborhood-based ones. One of the most common strategies in model-based collaborative filtering is the factorization matrix. In the next subsection will be presented a general vision of the neighborhood-based and model-based approaches.

2.1.2.1 Neighborhood-based Collaborative Filtering

The main idea of neighborhood-based filtering is the user might be interested in items other similar users also had interest in the past. Such approach can rely either on most similar users or most similar items information to recommend new items; both strategies are better explained in the next subsections.

2.1.2.1.1 User-based Nearest Neighbor Recommendation

This is one of the earliest strategies of recommendation, the main idea is to find similar users to the current (active) user. Such similarity is based in similar ratings the users have assigned to items in the past. Once discovered the most similar users to the current user, the rating to an unseen item *p*, the current user might be interested in, is calculated by taking the ratings the similar users have assigned to *p*. Two assumptions are made in this strategy: (i) if the users have similar tastes in the past they will have similar tastes in the future; and (ii) the user preferences are stable over time.

Consider $U = \{u_1, ..., u_n\}$ being the set of the users, $P = \{p_1, ..., p_m\}$ the set of products (items), and *R* a rating matrix *n x m* where $r_{i,j}$ is the rating the user *i* has assigned to item *j*, with $i \in 1 \ldots n,\ j \in 1 \ldots m$. The ratings are values from 1 (strongly dislike) to 5 (strongly like), if a user has not rated an item the correspondent entry remains empty. Taking such definitions imagine the task of predicting a rating a user has given to an item. Firstly, is necessary to calculate the similarity between users; then it is necessary to select the top-k similar users and finally calculate the predicted rating based in such top-k users rating.

To calculate two users' similarity a very common measure used in recommender systems is the Pearson's correlation coefficient. Where $\overline{r_a}$ is the average rating of the user *a*.

$$sim(a,b) = \frac{\sum_{p \in P}(r_{a,p} - \overline{r_a})(r_{b,p} - \overline{r_b})}{\sqrt{\sum_{p \in P}(r_{a,p} - \overline{r_a})^2 \sum_{p \in P}(r_{b,p} - \overline{r_b})^2}} \tag{4}$$

The results of Pearson correlation coefficient vary from +1 (strong positive correlation) to -1 (strong negative correlation). The reason for subtracting the average rating is to normalize the user behavior assignment of ratings, i.e., some users tend to never rate an item with a very low rating even when they dislike the item, their ratings tend to concentrate in higher values, and some users behave the opposite way. So, subtracting the average user rating normalize such rating assignment behavior.

The next step is the compute the prediction of rating for the target item, to compute the prediction of *a* to a product *p* it is used the following:

$$pred(a,p) = \overline{r_a} + \frac{\sum_{b \in N} sim(a,b) * (r_{b,p} - \overline{r_b})}{\sum_{b \in N} sim(a,b)} \tag{5}$$

Where *N* is the nearest neighbors of the user *a*. The choose of a set of nearest neighbors is one of the challenges to perform a good prediction. Two simple heuristics are to select only neighbors with positive similarity or to define a minimum similarity threshold. However, such heuristics can lead to the problems of selecting too much neighbors (influencing negatively the accuracy since it was took in consideration even users not comparable); or too little neighbors (what makes impossible to compute a good prediction). Empirical tests realized in the

Movielens database have shown that a good neighborhood size varies from 20 to 50 neighbors (HERLOCKER; KONSTAN; RIEDL, 2002).

2.1.2.1.2 Item-based nearest neighbor recommendation

Online retail stores have emerged and presented a new scenario to recommendation strategies where a huge number of users and items made impossible the computation of predictions in real time. Item-based nearest neighbor strategy has been successfully implanted in such systems because this strategy is better suitable for offline pre-processing what makes it possible to present almost real-time recommendations even for a very large rating matrix.

In item-based, differently from user-based strategies, an item rating is predicted by comparing other items ratings matrix. For instance, to predict a rating for an item the algorithm compares other items rating matrix and find the most similar items to the target one, then it takes the ratings the target user has assigned to the similar items and computes a weighted average of such items to assign as rating prediction to the target item.

To find the most similar items it is necessary a similarity measure, and for item-based strategy the standard measure is the cosine similarity. The metric measures the similarity between two n-dimensional vectors based in the angle between them. The similarity between a and b is get by:

$$sim(a,b) = \frac{\sum_{u \in U}(r_{u,a} - \overline{r_u})(r_{u,b} - \overline{r_u})}{\sqrt{\sum_{u \in U}(r_{u,a} - \overline{r_u})^2 \sum_{u \in U}(r_{u,b} - \overline{r_u})^2}} \qquad (6)$$

Where U is the set of users who have rated both a and b. As in the Pearson Correlation the mean user rating is subtracted to normalize the results. So, the similarity value will also vary from +1 (strong similarity) to -1 (weak similarity).

To compute the rating prediction, it is necessary to take the items that are similar to the target item, i.e. the item that we want to predict the rating, and the current user has already assigned ratings. Taking such similar items, the prediction of the user *u* rating to the product *p* is obtained by the weighted mean:

$$pred(u,p) = \frac{\sum_{i \in ratedItems(u)} sim(i,p) * r_{u,i}}{\sum_{i \in ratedItems(u)} sim(i,p)} \qquad (7)$$

2.1.2.2 Model-based Collaborative Filtering

Collaborative Filtering techniques have the characteristic of demanding only a set of ratings or usage behavior to provide recommendations to the users. There are basically two ways of collect such ratings or behavior, either by explicit or implicit feedback. System, like Netflix, implement the strategy of asking the user to explicitly provide the feedback by assigning a rating to the movies, such ratings are stars that vary on a scale from 1 to 5 stars. Other systems, like Amazon, can also implement an implicit feedback where users action like clicking on a product to obtain more information, the mouse movement, or buying a product can also be used to predict the user recommendations. In this subsection, however, we will present techniques to deal with explicit feedback.

In collaborative filtering two different entities must be related, the users and the items. To do so, the systems implement two main techniques: *the neighborhood approach* and *the latent factor models*. The neighborhood approaches were explained in the subsection 2.1.2.1. The latent factor models put both users and items in the same latent factor space, where the ratings are characterized by factors inferred automatically from user feedback.

To have more accurate CF methods it is necessary to go beyond the proposition of new modeling techniques. Looking for signals, or features, available in the data can reveal interesting aspects to improve the systems accuracy. An example is to look the items the user has chosen instead of the ratings given to such items to try to understand why the user has chosen such items and not others.

One of the most common techniques to model-based CF is the matrix factorization. It is relatively simply to implement and is able to handle big amounts of data. Such technique can also handle implicit feedback and temporal information.

For the rest of this section consider a rating matrix of m users and n items. We are considering also the letters u,v are assigned to refer to the users and i,j,l to the items. A rating given by the user u to the item i is indicated by r_{ui} and the predicted rating by $\hat{r}_{ui}$. The set of users who rated the item i is denoted by R(i) and R(u) is the set of items the user u has rated.

2.1.2.2.1 Simple baseline rating predictor

The CF models try to capture the interaction between the user and the items. Such interactions are in part responsible for the different ratings attribution. However, some of the different behavior in attribute the ratings are also explained by individual effects of the user and the item. For instance, some users tend to attribute higher ratings for all items they evaluate, as well as, some items tend to receive higher ratings independently of the user.

In (8) it is show a simple way to combine such individual effects (that do not consider user-item interaction) to produce a simple baseline rating predictor.

$$b_{ui} = \mu + b_u + b_i \tag{8}$$

Where b_{ui} is the baseline prediction for an unknown rating r_{ui}, and μ denotes the overall items average rating, b_u and b_i are the deviations of user u and item i, respectively, from the average. To estimate the values of b_u and b_i we can solve the least squares problem:

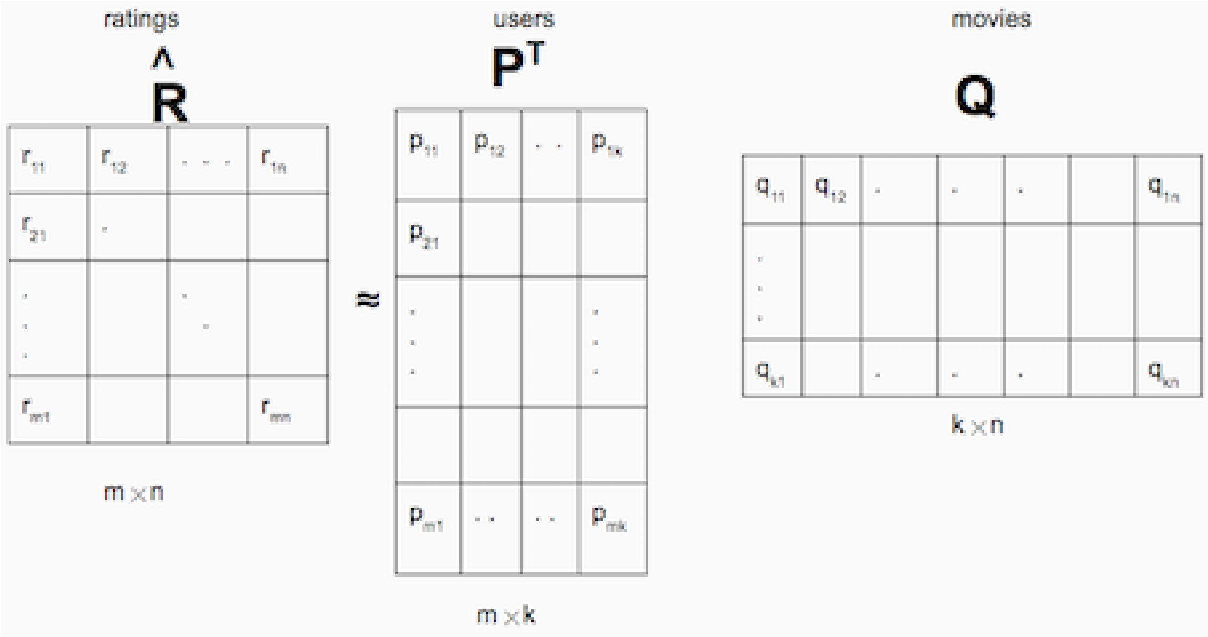
$$\min_{b_*} \sum_{(u,i)\in\mathcal{K}} (r_{ui} - \mu - b_u - b_i)^2 + \lambda_1 (\sum_u b_u^2 + \sum_i b_i^2) \tag{9}$$

In the equation $\mathcal{K}$ is the set of pairs (u,i) where r_{ui} is known. In the first part of the equation it is attempted to find b_u's and b_i's that fit the given ratings, and the second part avoids overfitting.

2.1.2.2.2 Matrix factorization models

Matrix factorization (MF) is one successful implementation of latent factor model (KOREN; BELL; VOLINSKY, 2009). In MF both the user and the items are characterized by vectors of latent factors, that are inferred from the rating patterns. The main idea of the technique is to take a rating matrix of m users and n items and split it in two smaller matrices Q and P. In a way that the dot product of $Q^T P$ could reflect with a minimum error the original rating matrix, with the advantage of filling the blank ratings. It is the same reasoning of splitting a number in its factors, for instance 10 could be split in 2x5 = 10. The Figure 2.1 shows such strategy of matrix split.

Figure 2.1 – The Rating Matrix Split Strategy



Source: (HALLER, 2016)

Each item i is associated with a vector q_i from Q also $q_i \in \mathbb{R}^f$; and each user u is associated with a vector p_u from P also $p_u \in \mathbb{R}^f$. For a given item i, the elements of q_i measure the extent the item has such factors, and for each user u, the elements of p_u measure the extent the user has interest in the items that are high in those factors, both values can be negative or positive. To capture the interaction between the item i and the user u, one must need solve the dot product between the vectors.

$$\hat{r}_{ui} = q_i^T p_u \tag{10}$$

Where $\hat{r}_{ui}$ is the rating prediction of the user u for the item i. The main challenge is to map the users and items to the vectors $q_i, p_u \epsilon \mathbb{R}^f$, and to do so, the algorithm must to learn the factors by minimizing the regularized squared error on the set of known ratings.

$$\min_{q^*, p^*} \sum_{(u,i)\in\mathcal{K}} (r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2) \tag{11}$$

Two approaches to minimize such error are *stochastic gradient descent* and *alternating least squares.*

2.1.3 Context-aware Recommendation

There are three approaches to include context in traditional recommender algorithms, the pre-filtering, the post-filtering and the model-based (ADOMAVICIUS; TUZHILIN, 2015). The biggest difference between these approaches is: while the first two utilize a filtering of data and then utilize a traditional recommender algorithm (i.e., an algorithm that consider only the user and the item information to suggest new items); the last approach modifies the algorithm itself to consider the context information during estimation of the recommended items.

The first approach is known as contextual pre-filtering because the algorithm makes a filtering in the data of all user ratings to select only those the user has given in a context of interest. Such filtered set of ratings will be used as input to a traditional recommender algorithm. A big challenge for this approach, though, is the need to define a context hierarchy to be used in the filtering phase. The hierarchy is necessary because it is not always productive to filter the data in the exact context of interest since it can cause a problem of sparsity, i.e. the algorithm does not have enough data to make a proper recommendation. Therefore, the utilization of a context hierarch can help the filtering in the task of selecting a more general context (located in a highest hierarchical level) to be utilized in the filtering, solving the sparsity problem. However, such context hierarchy demands a lot of domain knowledge to be built, as well as, the definition of the best granularity level is not an easy task.

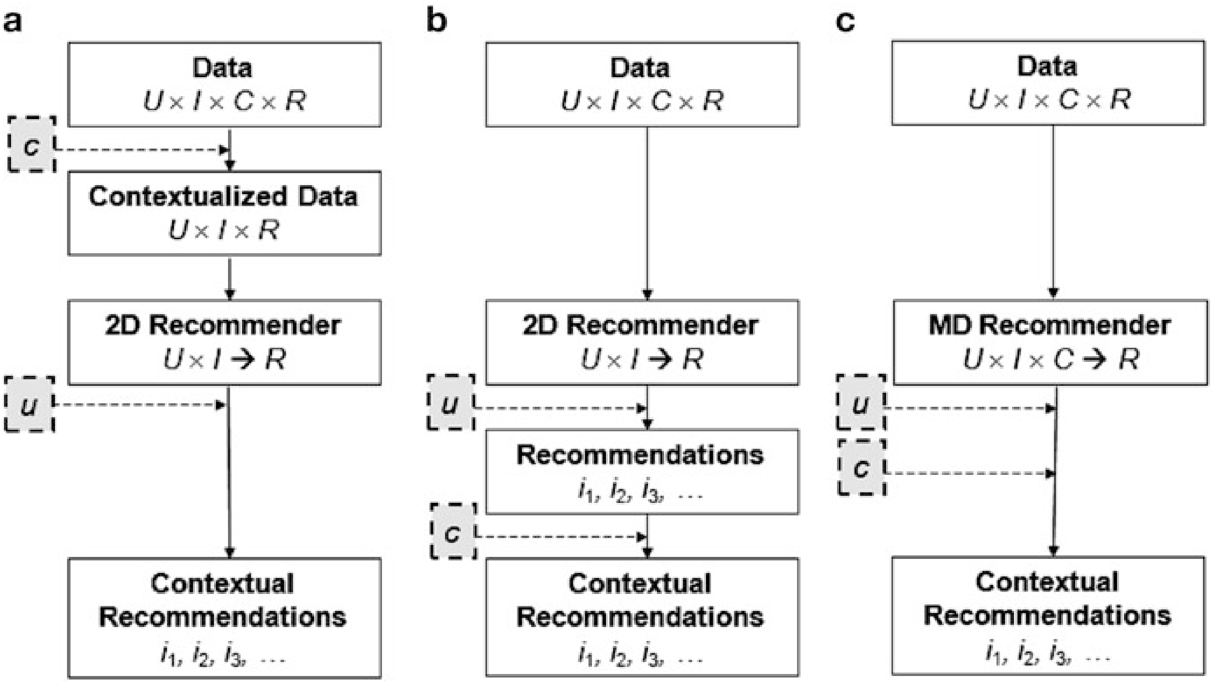
The second approach, the contextual post-filtering, has some similarities with the first approach in the meaning the algorithm realizes a filtering and applies also a traditional recommendation algorithm. However, in this approach, the filtering is realized after the recommendation is estimated. Other characteristic of this approach is the selection of only the items that satisfy the filter conditions (filtering) is not the default action performed after the estimation of the recommendation. To get the recommendation, the algorithm ignores the context information and behavior the same way it would a traditional recommender with non-contextual data. The post-filtering then either execute a filter over the items to consider only those which belongs to the target context or perform a re-ranking of all the items to put in the top-ranking positions the items of the target context, but it does not exclude the other items which belong to other contexts, only appearing in last positions on the ranking. The estimation of the recommendation can be performed either heuristically or probability. The main difference is the probabilistic use predictive models to estimate the probability of choosing an item in a target context, while the heuristic uses the patterns learned from the user data to provide the filtering.

A challenge to perform the post-filtering is related to the discover of user behavior patterns in the context of interest. Such patterns, discovered from the user data usage, are used to filter the recommended data. For instance, if a pattern discovered, in a movie recommendation system, the user watches only comedies in the weekends (such pattern can be discovered either heuristically or probability), then the filter will select preferably comedies for the weekends.

The third, and final approach, is known as the contextual modeling. Differently from the aforementioned approaches, this is the only one that really takes in consideration the context information during the estimation of the user rating. One of the ways to do so, is to use the Euclidian distance considering the user, the item, and the context instead of the user and the item only. This approach however presents one of the biggest challenges since it modifies the rating estimation function.

In Figure 2.2 is presented the three approaches for context-aware recommendation where ‘c’ represents the context and ‘u’ represents the user. In (a) is presented the pre-filtering where the data is contextually filtered ‘c’ before the application of the traditional recommender algorithm (2D Recommender), then the model is queried to show recommendations to the target user ‘u’. In (b) is presented the post-filtering where the contextual filter ‘c’ is applied after the 2D Recommender. Finally, (c) presents the process of contextual modeling where the algorithm of recommendation is able to use the contextual information ‘c’ directly in the estimation of the recommendation.

Figure 2.2 – Approaches for context-aware recommendation



Source: (ADOMAVICIUS; TUZHILIN, 2015)

Empirical results presented in (ADOMAVICIUS; TUZHILIN, 2011) show that the choice of the strategy of recommendation depends on the application developed but in general case post-filtering re-ranking strategy dominates the exact pre-filtering, which in turn dominates the post-filtering filter strategy.

2.2 Adaptive Hypermedia Systems (AHS)

Adaptive hypermedia systems is a research area that received much attention after the year of 1996 with the popularization of the Web. At the time, the area already carried some experience in the adaptation of hypermedia and the Web community received well such knowledge.

Some kinds of AHS ended to be very popular and received much of the research effort, they are: *educational hypermedia*, *on-line information systems*, *and information retrieval hypermedia*. The AHS can adapt either *content*, *presentation*, or the *navigation path* in between documents (BRUSILOVSKY, 2001).

2.2.1 Adapt to what?

One of the keys to proceed the adaptation is the user model, that supports the identification of the user and her needs of adaptations. In (BRUSILOVSKY; MILLÁN, 2007) it is presented an analysis of the user model in three perspectives; what to model, how to model, and how to maintain the model.

The authors present six useful features to model the user in an AHS:

- Knowledge: it is the most important user feature, it can reflect the user knowledge in a subject or a domain. This information can be used to adapt the navigation and the presentation of a system. Knowledge is a dynamic information, i.e., it can increase and decrease. One of the simplest ways to model user knowledge is to define a scale of quantitative values (for instance from 1 to 5) that measures the level of user expertise in the domain. Such values can be obtained from a self-evaluation or an objective test, for example. The user is then characterized in a defined category (e.g., novice, intermediate, expert). One of the shortcomings of this scalar strategy is the generalization of the user in a category, but the user knowledge differs from one subject to another, a general user can be an expert in text editing but a novice in formula editing, for instance. An alternative to overcome such problems is to use a structural model, that divides the body of knowledge into various fragments and evaluate the user knowledge to each fragment. The most popular form of structural model is an overlay model. In an overlay model the user knowledge is represented as a subset of the domain fragments. For each fragment the user knowledge is compared to which degree it relates to an expert knowledge. For instance, if in text editing an expert should reach the level 5 of knowledge but the current user only reaches the level 3. Other less used model is bug model that models also the user misconceptions to provide richer adaptation, but such model is difficult to implement and rarely used.
- Interests: this feature has been competing with knowledge to become the most important feature of an AHS. Popular also between the recommender systems, interests have been explored much more outside of the educational domain, where the user knowledge plays a more important role. Systems like

encyclopedias, electronic stores, museum guides, and information kiosks used firstly the user interest as feature of adaptation. More recently educational hypermedia also has been incorporating user interests as an important category. The first systems to model user interests used to put it into a weighted keyword vector space, this model become popular and it is still broadly used in the domains of information retrieval and recommender systems. Differently, in the adaptation domain the interests are modeled in a concept -level instead of a keyword-level. The interests are represented as a weighed overlay of a concept-level domain model. This is very similar to the modeling strategy of knowledge features as an overlay. Such concept model allows richer levels of personalization than the keywords ones. An adaptive museum system can separately model interests in the designer, style, or origin of a jewelry item, for instance. Other advantage is the links between the concepts in the model that helps to avoid sparsity a common problem in large overlay models. A drawback to the use of concept-level models is the need to manual annotation of the concept in each resource that is why such models are used generally in closed corpus systems (such as AHS). On the other hand, open corpus systems (such as information retrieval and information filtering) use a keyword level model but have the possibility to deal with much more quantity of information.

- Goals and Tasks: it represents the user immediate purpose for use with the adaptive system, it can be even the goal of the work, an information need, or a learning goal. The user goal is the most changeable feature, a user can change the goal from one session to another or can change her goal even in the same section more than once. Adaptation to user *work goal* was firstly done by adaptive interfaces and intelligent help systems, *learning goal* was explored by instructional planning and sequencing systems, and *information need* by adaptive information retrieval systems. Goal modelling can be done by simply using a catalog of predefined goals the system can recognize. Other more sophisticate strategy to model the user goals is to use a hierarch where in any time of the work the user will be at one specific level of the hierarch with one single goal. Such identification of goal then uses the rules associated with the recognized goal to provide adaptation. The user goal identification is a challenge to AHS, some systems ask directly the user of her goals and then try to learn to adapt to a completely new goal introduced by the user. Other strategies used recently is the user task identification through data mining algorithms, providing the user task-level adaptation.

- Background: it represents the user's experience outside the core domain of the application. The core domain means the main subject and interests of an adaptive application. Examples of background taking into account is a system that considers the user knowledge in medical language and subcategorizes the users by such knowledge, then presents the same content by using the medical terminology or everyday language. Other suggestions are to take in consideration the user job or language skills to adapt the system content. Background also can be used for presentation and navigation support adaptation, though its more common use is for content adaptation. Because the user background is a knowledge outside of the core domain, it is generally modeled by a stereotype model (an approach that attempt to cluster the users into several groups, then the adaptation is executed for such groups instead of adapt to each user individually). Since it is nearly impossible to deduce the background from the user's actions with the system it is usually informed manually by the user or a superior (e.g. a teacher).
- Individual Traits: individual traits are the set of characteristics that define the user as an individual. They are the personality traits, cognitive styles, cognitive factors, and learning styles. Such characteristics must be extracted by psychological tests and they are stable in time or if they change, they take a long period to do so. Cognitive and learning styles are the most used traits in adaptation. Cognitive style is an individual preferred way to organizing and representing information. This trait is used generally to provide adaptive navigation support, by identifying the user as field-independent, for instance, the user receive access to the navigation menu to explore the system as they prefer, and if classified as field-dependent, the user will receive the content sequentially with a map and a path indicator. Learning styles, is the way people prefer to learn, it is narrow than cognitive styles because it focus on human learning only. The adaptation focused on learning styles focus on match educational content with the identified style. Both cognitive and learning styles adaptation techniques are still imprecise in their results and their methods, there is no standardization in the mapping for a style and a method of adaptation.

- Context of Work: The increasing interest in ubiquitous and mobile computing has made the vision of context broader in AHS. Features like user location, physical environment and social context became of interest. The context and the user modeling are interconnect in a way that some features sometimes are common to both models. Some features that belongs to context model in AHS are: (i)user platform- because users of the same server-side use different platforms it is necessary to adapt the visualization (e.g. a version of a web page for mobile and a version for desktop). Other types of adaptation are to bandwidth, available software and hardware. Such adaptation is executed generally through stereotype models, because it is not practical to provide adaptation to each combination of features the user platform has. For instance, if the user access the system from a mobile device, instead of looking which device it is, the system can easily provide a version of the page adapted for small screens. If the device does not have a color screen or has a poor bandwidth, the system can provide pictures in black-and-white or low resolution; (ii)user location- it is naturally used in mobile context-adaptive. In this case the goal is not intended to fire adaptive presentation rules, but instead to determine a small subject of nearby objects of interest. It is of interest to systems like museum guides, tourist guides, and marine information systems; (iii)a broader view of the context- despite there is no definitive agreement about what features are context, in the mobile and ubiquitous computing there is a common core that includes *environment* and *human* dimensions. The environment dimension includes spatio-temporal and physical conditions (e.g. light, temperature, acceleration). The human dimension includes personal context (e.g. user pulse, blood pressure, mood), social context and user task. The consideration of the user task as part of the context and not the user model can be confusing. This happens because the research on context modeling happens in two different points of view. The user-centered view does not consider the task as context information while the device is. In the device-centered a range of user characteristics is considered as context information. To define a border between user and context modeling is important to observe that user modeling is focused on long-term user features, while context represents current features of the user and the environment.

In the next subsection, we will present some of the propositions for AHS in educational and information retrieval domains, since those are most correlated with this thesis interest.

2.2.2 How to provide adaptation?

In this subsection, it is presented the domains where adaptive hypermedia has been successfully employed and how such domains have been using adaptation in their systems. The focus is to present some techniques used in the domains of educational and information retrieval hypermedia.

2.2.2.1 Educational Hypermedia

One of the information educational hypermedia utilizes to execute adaptation is the user learning style (collected generally through psychological tests) or the user cumulated knowledge (collected automatically through the user interaction with the system). The cumulated knowledge was more utilized because it resulted in more accurate ways of adapting, differently from utilizing the learning style that resulted in a stereotyping of the user and does not lead to good results.

In (DE BRA, 2008) it is discussed many ways to provide adaptation in educational hypermedia. The author takes the three possible ways to adapt (content, presentation, navigation) and discuss what need to be taken in consideration to accomplish each level of adaptation. To adapt a hypermedia taking in consideration the user cumulative knowledge, for instance, the author considers two ways to present the same subject:

1- The creation of two versions of a same page where if the learner is familiar with all required concepts the subject is explained directly; and if the learner lacks some concepts the system inserts explanations of the concepts or provide a more introductory version of the content.
2- The system checks anticipated the knowledge requirements for the understanding of the concepts and if the current user does not possess such knowledge (verified through the learning path) it can hide, disable, or annotate the link accordingly.

The paper also presents some ways to capture user knowledge, it argues that some alternatives could be verify the user interaction with a page, or the application of a multiple-choice test.

In (MARILZA PERNAS FLEISCHMANN, 2012) the author proposes an ontology network that supports the modeling of the user learning style, the context, the device, and the learning objects of an e-learning environment. Such ontology is then instantiated and integrated in a e-learning system.

The author proposes the adaptation of learning paths by giving the user two possible options to navigate in the discipline content: “Tutorial”, that takes in consideration the knowledge pre-requirements defined by the professor, and “Livre” that shows all the discipline content without taking the pre-requirements.

It is also provided an adaptation of the presentation taking in consideration the device utilized to access the system, differentiating a cell phone from a desktop, for instance. Such adaptation processes are basically supported by the ontology and the user current situation, that in the work is defined as a context interval, for instance, if the user stayed in a classroom from 9 a.m. to 10 a.m. accessing the system through a smartphone, her situation identified was “in the classroom with the smartphone”. Such information is utilized to tailor the kind of adaptation that will be provided by using the ontology network as support.

Other interesting work in educational hypermedia is done by (BRUSILOVSKY et al., 2016), where the authors develop and evaluate the influence of using Open Social Student Modeling (OSSM) as a technique to enhance learning. OSSM is an evolution of Open Student Model (OSM) which is a technique that makes the user aware of her knowledge stored in the profile. Instead of only using the profile to provide personalization, an OSM system use it to increase the learner motivation by showing her progress in mastering a subject through a “skillometer”, for instance.

OSSM systems, on the other hand, makes the learner aware of other learners’ knowledges. So, the system shows the learner knowledge self-improvement and the progress of the other learners. In (BRUSILOVSKY et al., 2016) it is done by using a comparative grid, but it can also be done by a gamification leaderboard. The authors concluded the OSSM strategy enhanced learning, especially for students with lower prior knowledge.

In earlier approaches the model was used as a traditional tool for tailoring the system behavior as in (DE BRA, 2008), then in (MARILZA PERNAS FLEISCHMANN, 2012) such model besides adaptation becomes an ontology, this provides the model components to be described semantically, also present their relationships directly inside the model. In the last paper, the concern is not about the way to model the learner information but how the awareness of the model information influences the learn. This shows the central role of the learner model to support the adaptation strategies and how such strategies are related with each model.

In (DE BRA, 2008) the adaptation is focused in content and navigation support, as in (MARILZA PERNAS FLEISCHMANN, 2012) the strategies are concentrated in navigation support and presentation, finally in (BRUSILOVSKY et al., 2016) the main concern is in adapting the presentation.

2.2.2.2 Hypermedia for Information Retrieval

This type of adaptive systems became very popular after the web advent. The most challenging problem is to support the user retrieval activity in the Web unrestricted hyperspace (BRUSILOVSKY, 2001). Those systems can be classified also as *search-oriented* or *browsing oriented*.

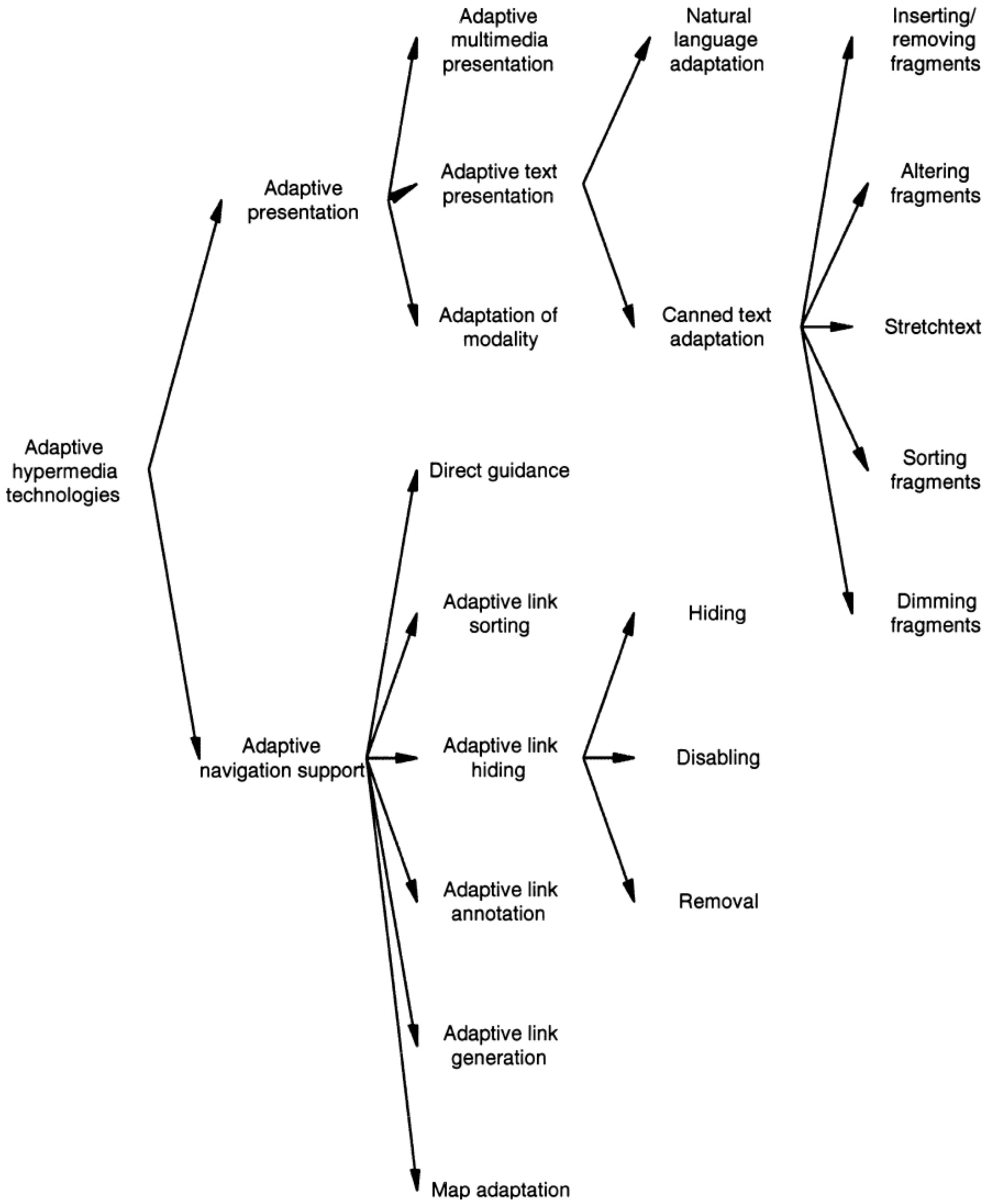
In the case of search-oriented the main goal is to provide links to documents that satisfy the user current information request. The main difference to traditional "one-shot" search engines is the consideration of user interests and preferences. Such consideration can be used, for instance, to remove links that will not be interesting or even to provide link annotation. For browsing-oriented systems the main goal is to support the users in the search-driven browsing.

To accomplish such goal the system can provide adaptive navigation by marking the links the user can have interest in, by providing visual cues to help the user decision, or even by suggesting links that will not be accessible from the current web page. This last functionality is supposed to be performed by a class of systems referred as *adaptive recommender systems* (BRUSILOVSKY, 2001). For the author, the definition of an adaptive recommender would be an algorithm used to suggest hyperlinks in an adaptive hypermedia.

2.2.2.3 Taxonomy of Adaptive Hypermedia Technologies

In Figure 2.3 is presented the main techniques used to provide adaptation of presentation and navigation support in general AHS. Such figure complements the techniques already discussed in the educational and information retrieval domains.

Figure 2.3 – Taxonomy of adaptation techniques



Source: Brusilovsky (2001, p. 100)

3 A LITERATURE MAPPING ON ADAPTIVE RECOMMENDATION

Systematic mapping study is a research method that gives guidelines to conduct literature reviews. It allows to make evidence given a domain to be presented at a high level of granularity and the identification of domain knowledge through clusters (KEELE, 2007).

The study presented in this work was conducted by following the guidelines proposed by Petersen et al.(PETERSEN; VAKKALANKA; KUZNIARZ, 2015). The essential process steps of the systematic mapping study are (i) definition of research topics; (ii) conducting the search for relevant papers (primary studies); (iii) screening of papers (inclusion and exclusion); and (iv) data extraction and mapping.

The Research Topics (RT) considered in the systematic mapping are:

- RT1. “Is there any recommendation approach to be applied in ubiquitous environments that is context-aware and adaptive?”
- RT2. “What are the most utilized techniques in each dimension of research?”
- RT3. “Which methods are most utilized to validate the proposed approaches?”
- RT4. “Which domains of problems the papers are related to?”

In (RT1), context-aware means the approach is able to manage the context information, adaptive concerning the approach adapts at least one of the characteristics defined by Brusilovsky and Maybury (MAYBURY; BRUSILOVSKY, 2002) or adapt the way of interaction with the user in a ubiquitous environment, for instance sending a visual alert through the TV or a sound in the radio. The (RT2) has the goal of identify the most common techniques utilized by the revised papers in the domains (dimensions of research) of context-awareness, recommendation, and adaptive systems. The (RT3) is related to the techniques utilized by the papers to classify their proposals. Finally, the (RT4) tries to identify which domains have been receiving more attention regarding the application of the approaches.

Such questions were defined based on the experience of the authors within the area of recommender, adaptive and ubiquitous systems. Their goal is to understand how the knowledge areas presented in (RT1) are related and to identify common strategies and even research gaps in the direction of turning recommender systems more adaptive.

The first step towards the answer of the research topics was the definition of a search string. To define it was identified the primary words in the research topic (RT1), such words were related to an area of knowledge or to the outcome of the researches. After identified, the primary words were submitted into a synonym dictionary in order to add the string possible synonyms and variations. Finally, a stemming was applied into the word set, to group similar radicals, and the string submitted to the search engines was:

> (*approach OR method* OR model*) **AND** (*recommend**) **AND** (*pervasi* OR ubiquitous*) **AND** (*aware OR sensi**) **AND** (*context OR situation*) **AND** (*adapt**).

The asterisk "*" is a wildcard character used to represent one or more characters (e.g. plural, variations of a word).

After defining the search string, it was submitted to four academic databases and search engines: ACM Digital Library, IEEE Xplore Digital Library, Springer Link and Elsevier. Each search engine uses different mechanisms and standards, then, the structure of the developed search string was adapted to apply it into each engine and conduct the search.

The inclusion and exclusion criteria are used to select relevant papers and exclude the ones that are not relevant to answer the research topic. The inclusion (I) and exclusion (E) criteria are:

- I1. Scientific Papers from conferences (e.g., conferences, congress, symposiums, workshops, etc.) or journals related to Computer Science;
- I2. Primary studies;
- I3. Full papers (four or more pages);
- I4. Papers from 2010 to 2016;
- E1. Papers in languages other than English;
- E2. Papers that do not discuss at least 3 of the main topics (ubiquitous, context-aware, adaptive or recommender systems);

The search retrieved 438 papers. Then, by the analysis of title, keywords and abstract we filtered the papers, to get those papers related to the research topic. Because of this first filtering 1st Filt. we have selected 173 papers.

Each one of these selected papers were analyzed by the reading of their title, keywords, abstract, conclusion and in some cases the section where the proposal is explained. The intention of this step was to understand the problem and the solution proposed by the paper. In this phase, we did not try to understand details of implementation, but we tried to understand in a more high-level the manner the solutions propose an answer to the presented problem. Each one of these papers were classified by the four areas related to the research as one of this mapping goals is to identify common strategies towards more adaptive recommendation and opportunities of research between the areas, we selected only the papers that were related with at least three of the research areas.

After this second filtering 2nd Filt., we got 57 papers that were clustered, read and analyzed. The results of our analysis showed that all 57 papers were context-aware, 42 have a recommendation approach, 55 were related to ubiquitous computing and only 5 of them presented an adaptation approach.

Table 3.1 – Search engines where the string was submitted

Search Engine	**Returned**	**1st Filt.**	**2nd Filt.**
ACM Digital Library	215	75	28
Elsevier	38	26	12
IEEE Xplore Digital	90	26	7
Springer Link	95	46	10
Total	**438**	**173**	**57**

Source: The author

3.1 Discussion and Results

This section is structured as follows. Subsection 3.1.1 describes an experiment realized with the papers keywords to obtain a clear vision of the set of papers used in this systematic mapping of the literature. The remaining Subsections (3.1.2 to 3.1.5) are related to each one of the four RTs, where they were explored and analyzed to answer them.

3.1.1 Clustering by keywords

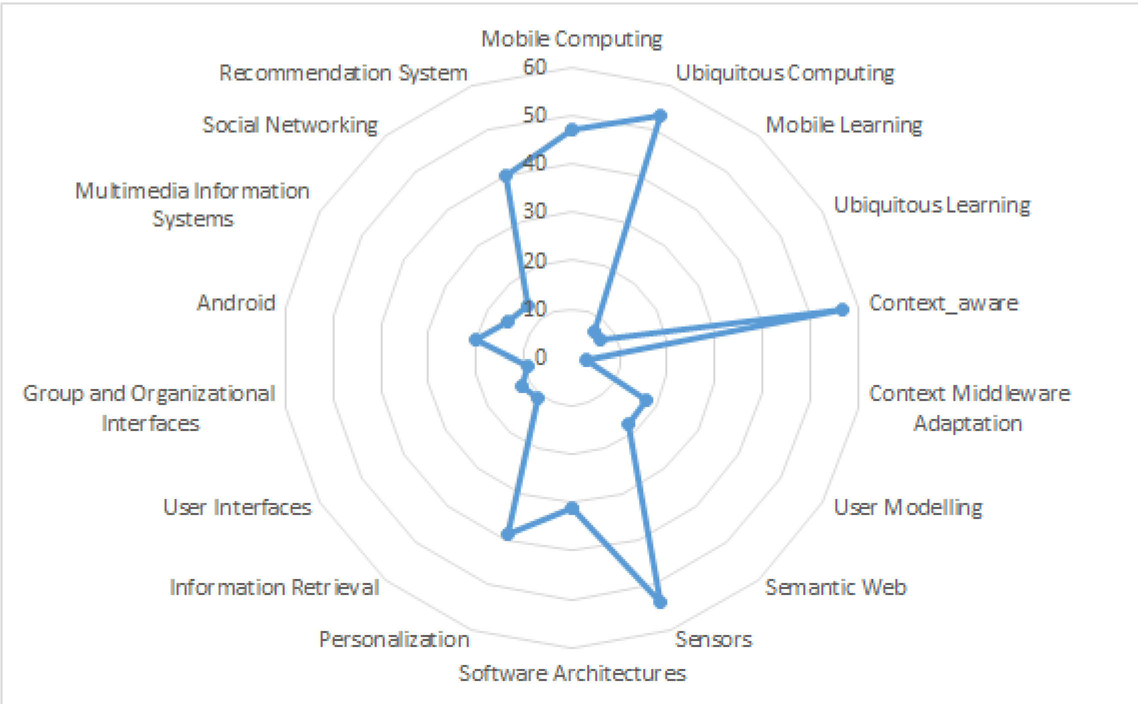
To have a better understanding and provide an overview of the main subjects of the paper set, the selected papers were grouped based on their keywords. The keywords were processed by the k-means algorithm to split the dataset in two clusters running 100 maximum optimization steps, this configuration gave the best results by creating the clusters. The goal of this processing is to characterize the papers dataset and understand how the subjects are related to each other. Such keywords were extracted from the pdf files and those that were considered too generic (e.g. algorithms, theory, experimentation) or appeared only once were manually removed. Considering of generic keywords could group papers under obvious or noninteresting characteristics, as well as, the keywords that only appeared once could disturb the clustering.

After this process, 21 representative keywords were selected. They were stemmed to avoid repetition of words; and by the end of this filtering, 18 representative keywords were employed to run the k-means algorithm.

The text of the papers was then analyzed again with the goal of identifying which of the representative keywords appeared as a subject in each paper. If a specific keyword does not appear explicitly in the keyword section of the paper or in the text, but the subject related to that keyword were treated in the paper the text analysis should capture such subject and mark such keyword as a subject of interest of the paper. Such analysis output is a dataset relating each paper with the 18 representative keywords. The dataset is available at http://bit.ly/2g6tAVt.

In Figure 3.0.1 it is presented a radar graphic that shows the distribution of the keywords in the paper set. As shown in the graphic, most of the works could be classified as related with the keywords of Context-aware, Sensors, and Ubiquitous Computing.

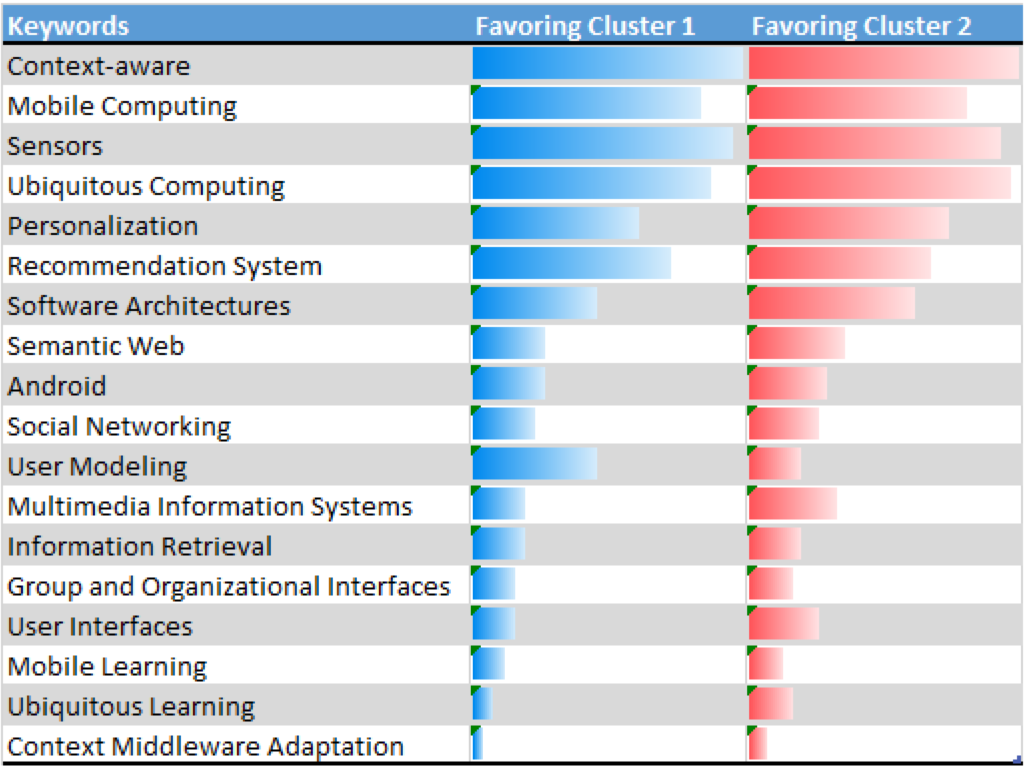
Figure 3.0.1 – The keywords distribution



Source: The author

The results presented by the algorithm are shown in Figure 3.2. The algorithm generated two clusters and despite the similar distribution of keywords in each cluster there are some of them that showed more discriminating in belonging to one of the clusters. The keywords *Multimedia Information Systems* and *User Interfaces* played an important role in discriminating the Cluster 2 as well as the keyword *User Modeling* presented importance in characterizing the Cluster 1. The fact of such keywords was not evenly distributed between the clusters shows that such characteristics are important to characterize not only the clusters but also shows that when working with multimedia systems it is important to spend some time planning the design of an interface, in this case the way to present the resources is a concern. This also shows that the user model is an important concern to the rest of the papers.

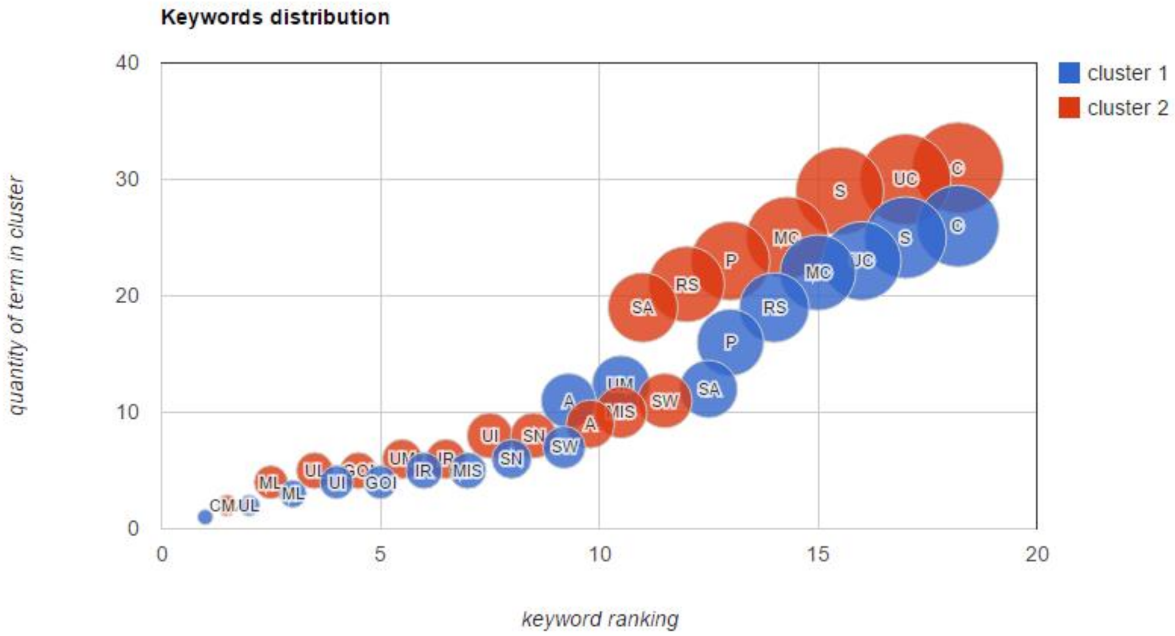
Figure 3.2 – The cluster composition



Source: The author

To better show the distribution of the keywords in each cluster, Figure 3.3, presents a bubble graphic that ranks each keyword based in its quantity inside each cluster. The keywords are presented in acronyms of their names. From the graphic, it is possible to note that the most frequent keywords in cluster 1 and 2 are Context-aware, Sensors and Ubiquitous Computing. This graphic also confirms the importance of user modeling to the cluster 1 and multimedia information systems, user interfaces and semantic web technologies to characterize cluster 2.

Figure 3.3 – Keywords distribution in each cluster



Source: The author

The clustering process were not able to point some concrete evidence about the tendency of having adaptive recommender approaches. However, the fact of presenting adaptive characteristics such as a concern to design of the user interface gives a clue about the existence of some approach that provides adaptation to recommender systems and this is exactly the subject of the first RT.

3.1.2 Research topic 1: existence of a hybrid approach

The first RT is related with the existence of a hybrid context-aware approach between the techniques of recommendation and adaptation in a ubiquitous environment. Since it was the main question of this literature review, an extra effort was applied in reading the papers and classifying them under the categories defined by previous works in recommendation and adaptation, those previous works were presented in Section 2. The domain of ubiquitous computing was not utilized to categorize the papers because in the RT the "ubiquitous environment" refers to the place where the approaches are applied; so, it is not viewed as an intrinsic characteristic of the approach. For that reason, all the presented approaches are related with a ubiquitous environment.

The results of the papers analysis are presented in Table 3.3 and the taxonomy utilized to construct it, is presented in Table 3.2. The taxonomy was structured based on the concepts presented in the works of the Section 2, such concepts provided a mean to organize and classify the contributions of each domain. Table 3.3 is organized in four groups of columns:

- i. General Information, contains generic information, about the paper, that are not related to the knowledge area;
- ii. Context-awareness, contains information about the manner the information of context is modeled, which level of information is taken, for instance, if sensor data is the main information utilized in the approach then the level is classified as LL (Low Level). Otherwise, if the approach manages information about the user situation, such as, the user is walking in the living room, then the level of information is considered HL (High Level). It also presents information about the goal to use contextual information.
- iii. Recommender Systems, contains information about the recommendation strategy, like, content-based, contextual or hybrid, as well as, information related to the algorithm strategy utilized to perform the recommendations, examples of strategies are: neighborhood-based, item-based and per-filtering.
- iv. Adaptive Systems, contains information about the target of the adaptation approach, such as, presentation, navigation style and content. It also presents information about the strategy utilized to adapt, such as, the adaptation is based on user preferences or it is based on the current context. If the context is used, for instance, one of the actions the system take is to adapt the system interface to the dimensions of the device screen, in that case the device characteristics is the context.

Table 3.2 – Taxonomy for papers classification

Dimension	Taxonomy	Acronym	Description
General Information	Type	CNF	Conference
		JN	Journal
		WSP	Workshop
Context Awareness	Modeling	GM	Graphic Model
		KV	Key Value
		LB	Logic Based
		O	Ontology
		OO	Object Oriented
	Level	HL	High Level
		IL	Intermediate Level
		LL	Low Level
	Usage	AS	Adaptation of Services
		DQ	Data Query
		IA	Interface Adaptation
		UR	Usage of Resources
Recommender Systems	Classification	CBR	Content-Based Recommendation
		CF	Collaborative Filtering
		CxBr	Context-Based Recommendation
		KBr	Knowledge-Based Recommendation
		H	Hybrid Approach
	Characteristic	IB	Item-Based CF
		MB	Model-Based CF
		NB	Neighborhood-Based CF
		PreF	Contextual Pre-Filtering
		PosF	Contextual Post-Filtering
		Mod	Contextual Modeling
		CsB	Case-Based
		CtB	Constraint-Based
Adaptive Systems	Factor	CS	Content Selection
		NPS	Navigation Support and Presentation
		P	Content Presentation
	Technique	BP	Based on Preferences
		BR	Based on Rules
		BC	Based on Context

Source: The author

Table 3.3 – Papers classification

General Information		**Context-awareness**			**Recommender Systems**		**Adaptive Systems**	
PaperRef	*Type*	*Modeling*	*Level*	*Usage*	*Classification*	*Characteristic*	*Factor*	*Technique*
(Abech, 2016)	JN	O	HL	AS	**	**	CS+P	BP
(Anacleto, 2014)	JN	GM	HL	DQ	CxBR+CBR	IB	**	**
(Arnaboldi, 2016)	JN	**	LL	DQ	H	NB	**	**
(Atif, 2015)	JN	O	HL	AS	**	**	NPS	BP
(Atif, 2014)	JN	O	HL	DQ	CxBR	IB	**	**
(Bagci, 2015)	CNF	**	LL	DQ	H	NB	**	**
(Beamon, 2010)	WSP	O	HL	AS	**	**	**	**
(Beer, 2013)	CNF	O	IL	UR	CF	NB	**	**
(Benouaret, 2015)	CNF	O	HL	AS	H	CBR+CF+PosF	**	**
(Biancalana, 2013)	JN	**	IL	UR	H	NB+Mod	**	**
(Böhmer, 2010)	CNF	**	LL	IA+DQ	CxBR	PreF	**	**
(Bourke, 2011)	CNF	**	LL	IA+DQ	CxBR	PreF	**	**
(BUCHANAN, 2010)	CNF	**	LL	DQ	CxBR	PreF	**	**
(Cheng, 2014)	CNF	GM	HL	AS	H	IB+Mod	**	**
(CHIN et al., 2013)	CNF	**	LL	UR	H	NB	**	**
(CHIN et al., 2012)	WSP	**	LL	UR	H	NB	**	**
(COLOMO-PALACIOS et al., 2016)	JN	O	HL	DQ	CBR	**	**	**
(CONSOLE et al., 2013)	JN	O	HL	DQ	CF	NB	**	**
(DORYAB; BARDRAM, 2011)	WSP	**	LL	IA+DQ	H	NB	**	**
(ELHAMDAOUI; ABIK; AJHOUN, 2011)	CNF	GM	HL	DQ	**	**	**	**
(EVERS et al., 2014)	JN	GM	HL	IA	**	**	NPS	BP
(FAN et al., 2015)	CNF	**	LL	DQ	CF	NB	**	**
(FRANCO et al., 2011)	JN	**	**	**	H	IB	**	**

(GUO et al., 2016)	JN	**	HL	DQ	H	PreF+NB	**	**
(HONOLD; SCHÜSSEL; WEBER, 2012)	CNF	LB	IL	IA	**	**	**	**
(HSU; HO, 2012)	JN	**	**	**	CBR	IB	**	**
(KAPTEIN; VAN HALTEREN, 2012)	JN	**	**	**	--	--	**	**
(KHALID et al., 2014)	JN	**	LL	DQ	H	NB	**	**
(KIRKHAM et al., 2013)	CNF	**	HL	UR	**	**	**	**
(KOEHLER et al., 2013)	CNF	**	LL	AS	CxBR	--	**	**
(LI; DU, 2012)	JN	GM	HL	DQ	H	NB	**	**
(LOQUES; SZTAJNBERG, 2010)	WSP	LB	HL	AS	**	**	**	**
(MAIA et al., 2012)	CNF	**	HL	UR	CF	NB	**	**
(MOEBERT; ZENDER; LUCKE, 2016)	JN	O	HL	AS	**	**	CS+P	RB
(NEVES; CARVALHO; RALHA, 2014)	JN	O	HL	DQ	CxBR	MB	**	**
(NOGUERA et al., 2012)	JN	**	LL	DQ	CF	IB	**	**
(OTEBOLAKU; ANDRADE, 2015)	JN	O	HL	AS	H	CsB	CS	BC
(PARATE et al., 2013)	CNF	**	IL	AS+IA	CxBR	--	**	**
(RIBEIRO; SANTOS; METRÔLHO, 2014)	JN	**	LL	DQ	CxBR	**	**	**
(RIENER et al., 2013)	JN	**	HL	UR	**	**	**	**
(RUOTSALO et al., 2013)	JN	O	HL	DQ+IA	H	IB	**	**

(SALMAN et al., 2015)	CNF	**	LL	DQ	CF	**	**	**
(SANTOS; RIBEIRO; METROLHO, 2012)	CNF	**	**	**	CBR	IB	**	**
(SCHAUB et al., 2012)	CNF	**	HL	AS	CxBR	--	**	**
(SCHEDL, 2013)	CNF	**	HL	AS	H	PreF+NB	**	**
(SCHEDL; BREITSCHOPF; IONESCU, 2014)	CNF	**	HL	AS	H	PreF+NB	**	**
(SHABIB; KROGSTIE, 2011)	CNF	**	LL	DQ	CxBR	--	**	**
(SHAO; GREENHALGH, 2010)	WSP	**	HL	UR	CxBR	--	**	**
(SMIRNOV et al., 2014)	JN	**	HL	DQ	CxBR	PreF	**	**
(SPEDALIERI et al., 2010)	WSP	O	LL	DQ+AS	CxBR	PreF	**	**
(WANG; ROSENBLUM; WANG, 2012)	CNF	**	HL	DQ+AS	H	IB+Mod	**	**
(WANG; WU, 2011)	JN	**	**	**	H	NB	**	**
(WOERNDL et al., 2011)	JN	KV	LL	--	**	**	**	**
(WONG; CHU; HAO, 2014)	JN	KV	LL	DQ+AS	CxBR	--	**	**

Source: The author

In Table 3.3, the papers that are not related with a concept had the cells filled with double asterisks (**), and the papers that present a different concept from those presented in the previous taxonomy had their cells filled with double hyphens (--). From the total of 57 papers, 3 of them (BELLAVISTA et al., 2012; LUCKE; RENSING, 2014; METTOURIS; PAPADOPOULOS, 2013) were of survey type and do not utilize the methods and techniques they present, they were not considered in the classification.

From the five papers that presented adaptive characteristics, only one of them also is also related to a recommender approach. The table also shows that adaptation is a recent topic being discussed in recommender domain, since the five papers that treat the subject date from 2014 to 2016. However, only one of the 57 papers present an approach that relates both knowledge areas of recommendation and adaptation, such fact answer the first research topic (RT) pointing that, in fact, there is an approach of recommendation that is also adaptive, but only one paper is not sufficient to draw consistent conclusions about the integration of adaptation inside recommender systems. It will take much effort exploring the area to reach a good maturity level.

In the only approach found to provide an adaptive recommendation (OTEBOLAKU; ANDRADE, 2015), the authors propose a module of software that takes an ontology and through some predefined rules make a post-recommendation adaptation of multimedia resources. This is done by verifying the user current context, the mobile device characteristics and the bandwidth quality, depending on the situation the resource can be summarized, shrinked or converted in another media format. However, even proposing such adaptation feature the authors do not validate the influence of such functionality in the whole recommendation process, they measure only the precision and recall of the hybrid recommender face a traditional one.

One of the causes for such small number of non-accuracy approaches is the challenge to find available data that contemplates all necessary information to validate such aspects. The option to such approaches is to conduct user studies, through the validations strategies of observational studies, case studies, experience reports or simulations. One of the downsides of conducting an observational study or an experience report is the dependency of finding a representative number of available users which interactions and answers could be used to measure the effectiveness of the novel approach. On the other hand, when executed case studies and simulation cannot be sufficient to evaluate all the implications of the approach on the user experience.

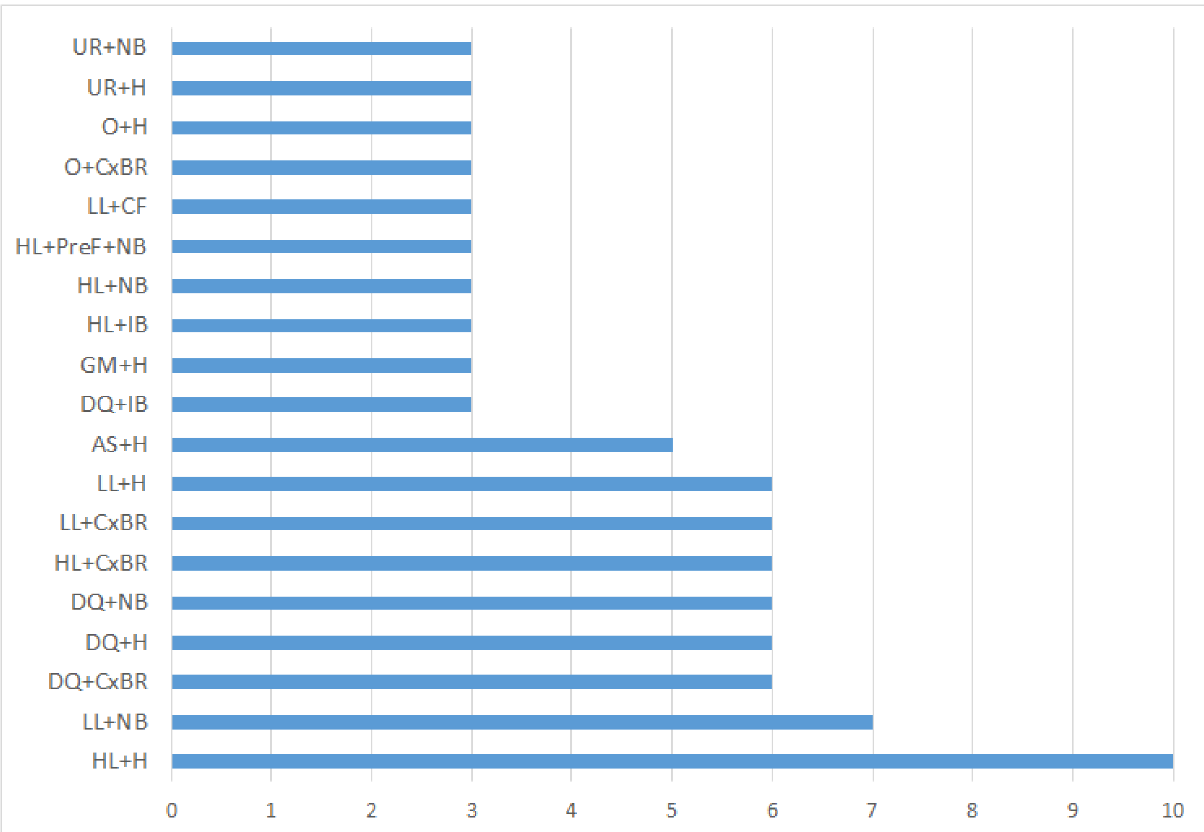
3.1.3 Research topic 2: most common techniques

To answer the RT2, an analysis was performed in the classified papers. The goal was to find the most common techniques utilized individually and the most common combination of techniques. Because of the reduced number of examples in the adaptive domain, the analysis of RT2 was only performed in the domain of Context-awareness and Recommender Systems. For adaptive domain the most used factor is *Navigation Support and Presentation (NPS)* and a combination of *Content Selection + Content Presentation (CS+P)*.

The results pointed out that Ontologies and Graphic Models are the most utilized techniques to model context, it also indicated that almost half of the papers deal with High Level context information and perform either Data Query or Adaptation Services as the main goal of context manipulation.

In the domain of Recommender Systems, most of the papers utilized a strategy based in Context or a Hybrid one, and the most common algorithm strategy are Neighborhood-Based, Item-Based and Contextual Pre-Filtering. Such data was extracted directly from Table 3.3.

Figure 3.4 – Most common techniques



Source: The author

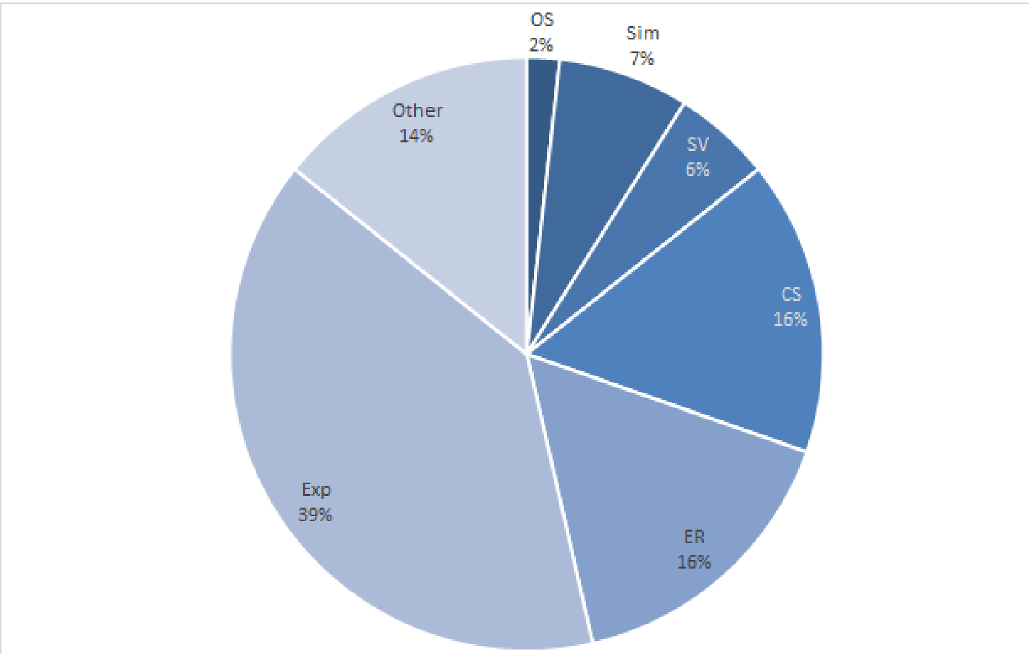
The data from the context and recommender domains were crossed to find those most common techniques utilized together. To perform such crossing, it was built a matrix of 16 lines and 15 columns where the techniques used together were counted and the results are shown through a bar chart in Figure 3.4.

The most common combination of techniques, are presented at the bottom of the graphic. The eight last bars are related with hybrid, context-based and neighborhood-based recommendation, the most common strategy combine a hybrid strategy of recommendation with high level context information. An interesting observation extracted from the last two bars is that generally when a hybrid recommendation approach is proposed it needs the context information in a high level of abstraction, on the other hand, approaches of recommendation that relies on more traditional methods as the neighborhood-based strategy generally uses the context information in a low level of abstraction.

3.1.4 Research topic 3: validation techniques

To answer RT3, it was executed a cataloging of the validation types realized by each approach in the set of papers. The classes of validation were partially based on the classes presented in (TONELLA et al., 2007) and partially based on the descriptions of the paper set.

Figure 3.5 – The validation techniques



Source: The authors

The classes of validation are: i) Observational Study (OS); ii) Simulation (Sim); iii) Survey (SV); iv) Case Study (CS); v) Experience Report (ER); vi) Experiment (Exp); vii) Other. Observational study is an unobtrusively method of gather observations to statistically support a hypothesis, often taken by a survey. Simulation, is the method when it is utilized a software to generate data with the intention to mock a behavior in real world, often observed in pre-tests of ubiquitous approaches. Survey, is not a method of validation, it is instead a tag for papers of literature review type. Case Study it is the treatment of one determined case, and the collection of data generated by the approach, the goal is to obtain insight in the attributes of a set of products or processes. Experience report is also the treatment applied to one specific case, but no particular effort is applied in controlling the context, the goal it is not to have an insight but to show superiority of the proposed approach. Experiment is is an execution applied under control to observe the effects, generally in domain of recommender system it is taken a dataset and the behavior of the system is measured and statistically validate. Other, is the tag utilized to characterize papers with very specific validation method which is not covered in this classification and also to characterize papers that do not provide a validation to their approaches.

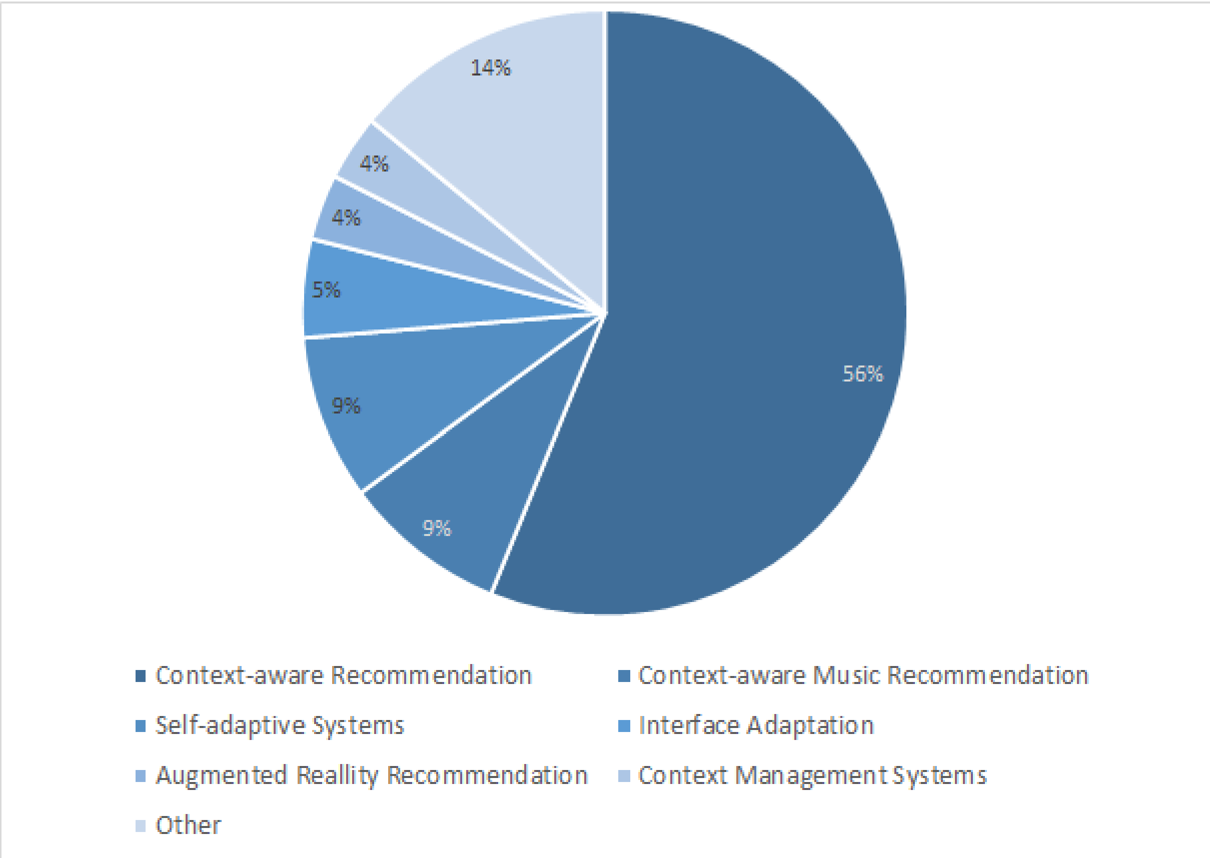
By the analysis of Figure 3.5, it is safe to say that the most popular strategies of validation are Experiment, Experience Report, and Case Study. One of the possible reasons for Case Study and Experience Report been two of the most popular methods of validation is because the complexity to gather raw information and process it up to get high level context information. Our analysis has shown a significant quantity of papers which proposes an approach, implements it but, because the difficulty to have input data and more difficult to process low level context into high level context information, such papers assume the preexistence of their high-level information input. This assumption limits the type of validation to be performed; so, the approaches are often validated through Case Studies and Experiences Reports.

3.1.5 Research topic 4: application domains

The RT4, is related with the domains of application of each paper. A granularity challenge appears in this phase, because each one of the 57 papers presented very specific domains of application. However, it was necessary to group such domains involved in each research in more general groups and for that reason the papers were divided in 6 domains of application: i) Context-aware Recommendation; ii) Context-aware Music Recommendation; iii) Self-adaptive Systems; iv) Interface Adaptation; v) Augmented Reality Recommendation; vi) Context Management Systems. The reason there are 3 domains related with recommendation is because the number of papers found in this domain was bigger than the others. So, we decided to split the domain of recommendation in 3 of the most popular within the papers.

In Figure 3.6 it is shown the distribution of the papers in the 6 identified domains and the papers with a domain that could not be included in one of these were classified as Other. More than half of the papers have applied their contributions in a specific area of context-aware recommendation, for instance, there were papers recommending food products in a market, recommending a break to the coffee and even people with the same interests in a conference.

Figure 3.6 – The application domains distribution



Source: The author

3.2 Conclusions of the Mapping

This chapter presents a systematic mapping of recommendation approaches used in ubiquitous environments. The intention was to screen the literature to find approaches that also propose adaptation features. The search for papers was conducted in four search engines and after two filtering, 57 papers were selected for analysis. Such papers were then clustered based on their keywords, such process helped the area characterization, but it was not sufficient to answer all the research topics proposed in this chapter.

The papers were classified by the set of 18 keywords and the clustering showed some research subjects were important to characterize the set in each cluster. Such separation of papers by subjects helped in the planning of the reading and analysis of the approaches, it also helped to extract some previous conclusion, such as, the papers that generally were doing some proposition with multimedia information systems also have a concern of plan the design of their user interface.

The results and analysis presented in Section 3.1 have contributed to identify the research area and lead us to draw some conclusions. The first is the existence of very few number of adaptive recommendation approaches even existing an increasing interest in more sophisticate recommending approaches. This mapping can also be characterized as an exploratory search (MARCHIONINI, 2006), because it digs into propositions of a relatively recent area trying to discover how are being done the design and evaluation of the propositions. The only paper found to be working simultaneously with recommendation and adaptation did not present an evaluation of their adaptive contribution, it rather focused on the evaluation of the accuracy of their approach.

So, one of the interpretations for such small number of adaptive recommendation approaches is the challenge to validate aspects not related with the accuracy of rating prediction in case of accuracy there is already a standardization of methods and metrics to evaluate it. When proposing an increase of user satisfaction, for instance, such proposals need to conduct user surveys and it is not easy to find a good number of available users.

It also has shown that a considerable amount of the papers validates their approaches utilizing case studies and experience reports, one of the reasons for it is because of the high level of contextual information utilized by the approaches and the lack of datasets to perform experiments in characteristics other than accuracy.

The papers studied in this review also helped to understand that a future adaptive recommendation approach will need to manage the context information in a high level, since this level of context make it possible the identification of situations. Such situations provide an important source of knowledge to the system, because it makes the system able to know, for instance, the user is not only in a determined location but what she is doing in that location. That way it is possible to provide recommendations more adapted to the user. If the system knows the user is in a museum it can provide recommendations about interesting art work, but if it knows the user is there to visit the temporary exposition, the system can adapt the path of the visit to lead the user directly to the exposition.

Besides high-level context, an adaptive recommendation approach will also need to know how to balance their recommendations in consideration with the user current situation. So, such approach will be built using a context-based or a hybrid recommendation strategy as already shown in most of the papers studied.

Another important conclusion this study leads us, is to recognize the need for adaptation is a demand related directly with the recommendation process independently of the ubiquitous domain. For this reason, in the rest of this text we will refer to adaptation of recommendation processes, ubiquitous or not.

4 AN APPROACH FOR ADAPTATION IN RECOMMENDER SYSTEMS

In this chapter it is presented an approach to provide adaptation features to a traditional recommender system. Such approach is inspired by the Adomavicius one to include context in the recommendation process. However, an extra step is demanded when adapting, it is the "item handling". The chapter also presents what distinguish adaptation from recommendation and how an adaptation strategy can be inserted in a recommendation process and the effects it would cause to the whole filtering process. An algorithm for insertion of profile characteristics in the rating prediction of a factorization matrix function is also proposed in this chapter. Then, in the last part of the chapter it is structured an approach to provide adaptation to a recommender system of educational resources. This last approach is an instantiation of the generic one defined.

4.1 The Adaptive Recommendation Process Overview

As already demonstrated before by Adomavicius et al. (ADOMAVICIUS et al., 2005) the awareness of the user context information can make recommender systems able to adapt their item-set and deliver better recommendations. The context, in their case, was identified as the information that is taken to change the results or to filter the initial item-set used by the traditional recommendation methods.

In the Adaptive Hypermedia domain, the context is also identified as one of the elements that supports the adaptation strategies (BRUSILOVSKY; MILLÁN, 2007). A whole set of adaptation strategies is context-aware, some examples are adaptation of navigation style for museums' audio guides (BENOUARET; LENNE, 2015), of content for learning systems (WANG; WU, 2011), or presentation for e-learning (PERNAS et al., 2012). In all the examples the authors used the user context information to tailor the system behavior.

Besides the context, both systems (recommender and adaptive) model the user profile information differently to take advantage of the user specificity and provide a personalized experience. In the case of recommender systems, the more common technique to build a profile is through the user interaction with the system, for instance rating a set of recommended items.

AHS however, also need to model information that cannot be collected directly from the user interaction with the system. Brusilovsky and Millán (BRUSILOVSKY; MILLÁN, 2007) depict some of such information as the user background, cognitive style, and affective state.

The combination of adaptation techniques inside a recommendation process it is not a trivial task, since the traditional recommender systems take in consideration only two dimensions to compute their recommendations. It raises some challenges, like: When the recommendation happens? When the adaptation happens? Which are the consequences when one happens before the other? Is it possible to merge both processes in one algorithm?

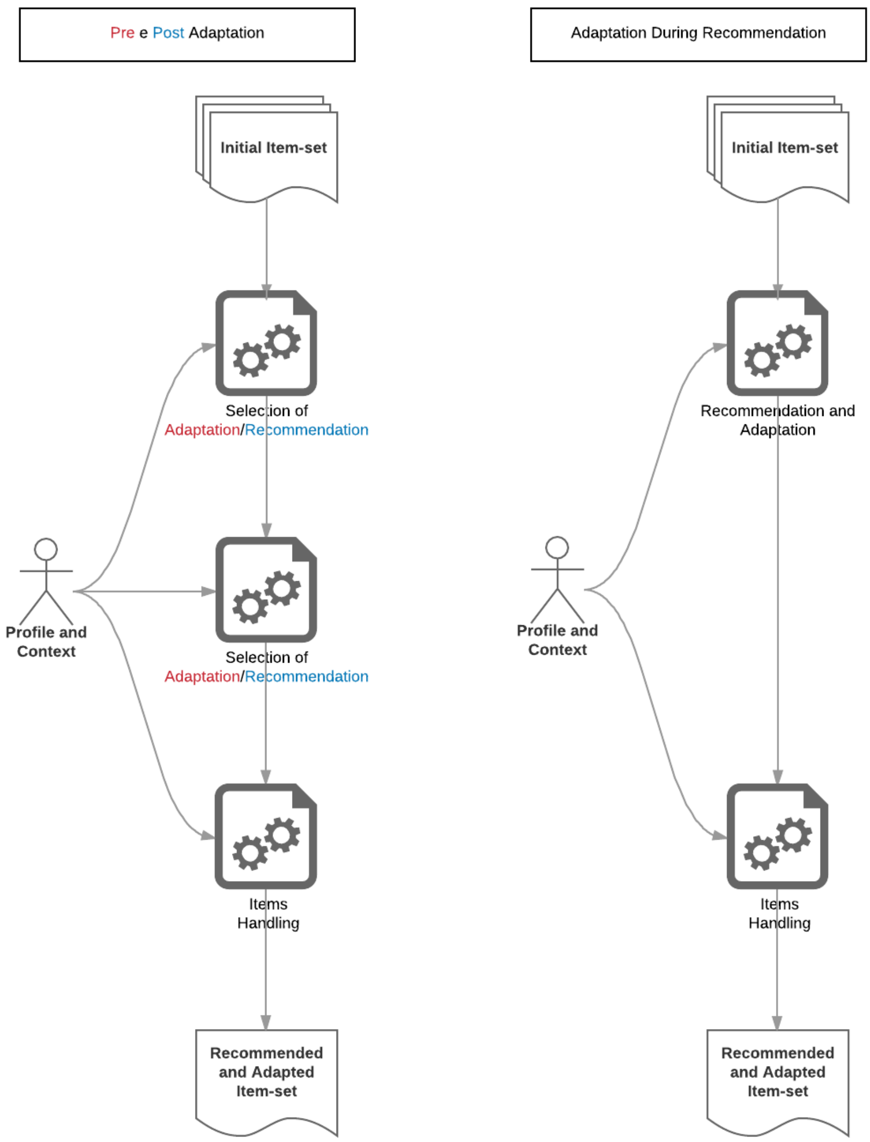
In Figure 4.1, it is shown an overview of how the strategies of recommendation and adaptation can be combined with the user context and profile information to support a process of adaptive recommendation. The adaptation can happen either before (pre-adaptation), after (post-adaptation), or during (modeling of adaptation) the execution of the recommendation algorithm. The processes perform four main activities “Selection of Recommendation”, “Selection of Adaptation”, “Recommendation and Adaptation”, and “Items Handling”. This last task is the only one shared by the three processes. Each process is better described in the next subsections.

4.1.1 Pre and Post Adaptation

These two processes are represented by the same (left) process in Figure 4.1. The main characteristic is the adaptation task can happen either before or after the estimation of the recommended set of items, defining two distinct processes, the “Pre-Adaptation” and the “Post-Adaptation” respectively. One of the advantages of such processes is the compatibility with the already existing recommendation algorithms, since the adaptation happens as an extra step of the recommendation process. So, it is not necessary to change the algorithm of item filtering used in the recommendations step.

When the adaptation happens before the recommendation it is characterized a process of Pre-Adaptation, and it is performed a pre-filtering on the item-set. This extra filtering will guarantee that only the items that accomplish the demands of the profile and the context will be used as input to the recommendation algorithm. For instance, there is a recommendation algorithm in an e-learning system, which suggest books to the students of a discipline. Such books are described by a metadata pattern like LOM(“IEEE Standard for Learning Object Metadata”, 2002) or OBAA (VICARI et al., 2010) and each student has her learning style collected through the form defined by Felder and Solomon (FELDER; SOLOMAN, 1991). So, in this case, a pre-filtering over the initial item-set will guarantee that only books that attends the student learning style (let's say visual) will be passed to the recommendation algorithm. After the filtering performed by the recommendation algorithm is done, another algorithm of item handling inspired in the techniques described in the classical taxonomy of Brusilovsky (BRUSILOVSKY, 2001) and other more recent techniques (BRUSILOVSKY et al., 2016; GASPARINI et al., 2010) change the recommended items. Such change comprises an adaptation of the book content, for instance highlighting the important sections based on the evolution of the student knowledge on the subject, it also suggests a learning path through the chapters emphasizing graphs and tables since the student has a more visual learning style.

Figure 4.1 – The strategies for adaptive recommendation



Source: The author

The process of Post-Adaptation is analogue to the Pre-Adaptation, with two important differences. First, the initial item-set is used directly as input to the recommendation algorithm and the adaptive filtering is performed after the recommendation one. Second, since the adaptive filtering is performed after the recommendation algorithm, it is not necessary to perform an item-set filtering. The adaptation algorithm can choose between filtering the items that does not accomplish the context and the profile demands, or it can choose to re-rank the recommended items, putting in the top-k positions the objects that present a good predicted rating and also are better adapted to the profile demands. The items handling is performed in the same way as defined in the Pre-Adaptation process.

One of the benefits of the Pre and Post-Adaptation processes is the compatibility with existing recommendation algorithms. A cold start problem can happen when the adaptive filtering is realized before the recommendation, because its filtering can decrease the number of neighbors and ratings to less than the minimum to perform well the recommendation. This problem does not happen in the Post-Adaptation process since the re-rank or post-filtering is realized after the recommendation task.

4.1.2 Adaptation During Recommendation

In the Adaptation During Recommendation process the user profile features are included within the recommendation algorithm. So, instead of realizing a Pre or a Post-Filtering over the items, the recommendation algorithm embodies a strategy to suggest interesting items that can also meet the characteristics demanded by the user profile.

One simple way to make the recommendation algorithm embodies such adaptation strategy is to put a bias in the computation of the recommended item-set. This bias would make the items that are more suitable to be interesting and to meet the other profile needs (e.g. more figures and maps to visual learning styles) to be put in best positions in the rank. The classic matrix-factorization algorithm (KOREN; BELL; VOLINSKY, 2009), for instance, can be adapted to put such bias in the rating prediction step.

As an example, imagine a system to recommend books to a learner who has a medium level of knowledge in the Object-Oriented Programming (OOP) subject. It is also known that other users who have a medium level of knowledge in OOP tend to rate the items belonging to such subject with 0.5 points less than their medium rate to other items. So, when the algorithm computes the estimated rating the user would give to a new item it would already take in consideration such bias. Each profile characteristic should contribute with a specific bias and the sum of such biases will influence in the rating prediction.

The papers of Baltrunas (BALTRUNAS; LUDWIG; RICCI, 2011; BALTRUNAS; RICCI, 2009) can be used as a starting point to the definition of an adaptation strategy. But instead of considering an average global user bias, in the rating prediction we take the sum of the user profile biases. The rating formula then becomes the following:

$$\hat{r}_{uiz_1 \ldots z_k} = q_i^T p_u + \bar{\imath} + \sum_{j=1}^{k} B_{ijz_j} + b_i \quad (12)$$

Where $\hat{r}_{uiz_1 \ldots z_k}$ is the rating the user u would give to item i, when the profile factors $z_1 \ldots z_k$ are considered. $\bar{\imath}$ is the average of the item i ratings, b_i is the bias of the rating given to item i when compared with the mean rating of the items in the system. B_{ijz_j} are the parameters modeling the influence of the profile factor and the user. A profile factor is a category of a profile information that can influence the rating the user give to an item, for instance a profile factor for visual learner is the "learning style". Being $z_j = 0,1, \ldots, c_j$, and $z_j = 0$ means the *j-th* profile factor is unknown, while the other values are the profile bias for such user. Where k is the number of profile factors. Let us define also that each profile factor can assume a number of possible values c_j. For instance, if a recommender system for educational movies ask for the user to inform 2 educational factors (Level of English, level of Math) $k = 2$ and if each factor can assume three different values (Low, Medium, High), each $c_j = 3$.

This definition allows to pick the level of granularity the profile information will be used to tailor the rating prediction. Using the influence of the profile factor (category) cause over a user behavior in her ratings, for instance, if a user who informed her level of history tends to give a rating higher than the average rating. Or it can be used also the profile information (value of each profile factor assumes to a user) to verify the behavior, for instance a user can have a low level of English (a value assumed by the profile factor Level of English) and how it influences her behavior in rating.

One of the advantages in using an algorithm of adaptation during the recommendation is the less possibility of happening the cold start problem, that can happen more frequently when a strategy of Pre-Adaptation is in use, since the algorithm receives less data to learn the user preferences.

4.1.3 Selecting an Adaptation Strategy

The selection of the adaptation strategy will depend on the system goal. For instance, if the goal is to provide recommendations adapted to the user current location, a simple context-aware recommendation can be used as a strategy. But if the user needs a list of objects adapted to the current context and to her needs of content, or navigation style, or presentation, then it is necessary to provide adaptation to the user profile needs during the recommendation.

Another influence on the selection of the adaptation strategy is the level of context data. It is easier for simpler recommender systems to process low-level context information than it is for a system that demands a user knowledge that goes beyond her preferences. It happens because adaptation generally needs a high-level user profile information to decide which strategy will be performed. For instance, it is necessary to know the user learning goals, levels of knowledge, and learning style, before recommending some learning resources. Such information hardly can come as a raw sensor data, it is necessary some processing over it to be used by an adaptive algorithm.

The adaptation strategy can take place before, during or after the estimation of the recommended item-set. One of the things that will influence the choice where the adaptation will take place in the process is the size of the item-set. Bigger item-sets are more suitable for Pre-Adaptation because it is less probable that way to happen the cold start problem, since there are more instances of rating available to learn the user preferences. Smaller item-sets are better suitable for Post-Adaptation for the opposite reason. The Adaptation During Recommendation method is also affected by cold start, mainly for new users and new items, but in this case the problem does not arise because of a filtering in the input data, like in the Pre-Adaptation, it is normal behavior for a collaborative filtering algorithm.

4.1.4 Algorithms to adaptive recommendation

There are two algorithms to provide adaptive recommendation used in this thesis. One to generate a list of recommendations adapted to the user profile, and other to generate a list of similar items, this last one is a classical algorithm for content-based recommendation and it is not a contribution of this work. Both algorithms are presented in this section, the algorithm for adaptive recommendation is presented in two parts being the first one (Figure 4.2) the loop where the recommendation is computed, and the second (Figure 4.3) is the procedure which computes the predicted rating a specific user would give to a specific item, taking in consideration the biases of item and profile categories. The last algorithm (Figure 4.4), presents the procedure which computes the list of similar items.

To define the list of items the user would like and also would be best adapted to the profile requirements it is defined a matrix-factorization algorithm. In this algorithm, shown in Figure 4.2 it is used a technique of stochastic approximation of the gradient descendent optimization (lines 5 and 6). Such approximation has the goal to minimize the error of rating prediction (line 4). The error is computed by the difference between the actual rating given by the user and the rating prediction, which is computed by the prediction function. The algorithm input is R which is the rating matrix nxm where n is the number of users and m is the number of items, each cell of the matrix contains a rating or is blank.

There are also two parameters which are γ that defines the algorithm learning rate, and λ that defines a regularization weight to avoid overfit. P and Q are matrices kxn and kxm respectively, where k is the number of latent factors used in the algorithm, such matrices are initiated with random values. The $number\_epochs$ is the number of iterations needed to the algorithm learn. Finally, the algorithm returns a matrix $\hat{R}$ of the size nxm containing the rating predictions for all the items available in the system. Differently from the R matrix, $\hat{R}$ is not sparse and all his cells are filled with rating values.

Figure 4.2 - Algorithm for adaptive recommendation

Algorithm 1 Compute the prediction matrix

```
1:  procedure ADAPTIVE-RECOMMENDATION(R, γ, λ)
2:      for epoch in number_epochs do
3:          for u, i in users and items do
4:              e = R[u, i] − prediction(U, I, P[:, u], Q[:, i])
5:              P[:, u]+ = γ ∗ (e ∗ Q[:, i] − λ ∗ P[:, u])
6:              Q[:, i]+ = γ ∗ (e ∗ P[:, u] − λ ∗ Q[:, i])
7:          end for
8:      end for
9:      R̂ = Q^T ∗ P
10:     Return R̂
11: end procedure
```

Source: The author

The prediction function is presented in Figure 4.3, the inputs for this algorithm is the matrices $P[:,u]$ and $Q[:,i]$, as well as two other matrices U and I, that contains the user profile and the item biases. The matrix U is nxz where n is the number of users the system has, and z

is the number of profile features defined by the system. So, if the system has 100 users and each of these users has informed 3 profile features, e.g. "level of English", "learning goal", and "cognitive style", the matrix U will have a column for each feature and each cell will present the bias value for the value the user has in that category. For instance, if a user i has a low "level of English", and by the dataset analysis is known such users tend to rate -0.2 points when compared to the mean rating, the position $U[i, 2]$ = -0.2; being 2 the column occupied by the "level of English" feature. The I matrix has a similar structure, but instead of n line (meaning the users) the matrix has m lines representing each item and one column containing such item bias when compared to the mean rating. Finally, the function returns the predicted rating which is given by the dot product of the P and Q matrices, plus the items mean rating, plus the sum of user profile biases, plus the item bias.

Figure 4.3 - The rating prediction function

Algorithm 2 Compute the rating prediction

```
procedure PREDICTION(U, I, P, Q)
    r̂ = Q^T * P + ī + ∑ b_u + b_i
    Return r̂
end procedure
```

Source: The author

The last algorithm, is the content-based recommendation (Figure 4.4) which computes the list of similar items. To implement it was used the scikit-learn package for python, so the algorithm is simpler than the presented before. The input is the items information, the first step is to compute the TF-IDF matrix, and finally it is computed the cosine similarity matrix, which is returned. In the prototype it was shown a list of the top-5 most similar items.

Figure 4.4 - The algorithm for similar items recommendation

Algorithm 3 Compute similar movies

```
procedure CONTENT-BASED RECOMMENDATION(movies)
    M = Tf-Idf(movies_header)
    Similar = cosine_similarity(M)
    Return Similar
end procedure
```

Source: The author

One of the most remarkable differences between the “adaptive-recommendation” and the “content-based recommendation” algorithms, is the absence of a user profile in the second. To compute the most similar items it is taken in consideration only the item characteristics independently of the user. This means that each different user will see the same list of similar items when looking to items details. The biggest advantage of this algorithm is its efficiency when compared to the matrix factorization one. The reason is because in the matrix factorization the user features must be inserted in the loop of the recommendation. However, for best performance, the results of the matrix factorization algorithm are saved in the database for instant access, being such results updated periodically.

4.2 Approach for Adaptive Recommendation of Educational Resources

As already discussed by (MANOUSELIS et al., 2011), provide recommendation to Technology Enhanced Learning (TEL) domain, is different than providing recommendation to other domains. An important feature to take in consideration when recommending to TEL is the user is now a learner and besides her preferences there is also a learning goal that needs to guide the recommendation.

The authors show in Figure 4.5 the comparison between a “Generic” or “Traditional” recommender system and a TEL recommender. In a nutshell, the biggest difference is the consideration of the learner knowledge (profile) and her learning goals.

Figure 4.5 - User tasks in TEL recommenders

Tasks	*Description*	*Generic recommender*	*TEL recommenders*	*New requirements*
Existing User Tasks supported by Recommender Systems				
1. ANNOTATION IN CONTEXT	Recommendations while user carries out other tasks	E.g. predicting how relevant the links are within a web page	E.g. predicting relevance/usefulness of items in the reading list of a course	Explore attributes for representing relevance/usefulness in a learning context
2. FIND GOOD ITEMS	Recommendations of suggested items	E.g. receiving list of web pages to visit	E.g. receiving a selected list of online educational resources around a topic	None
3. FIND ALL GOOD ITEMS	Recommendation of all relevant items	E.g. receiving a complete list of references on a topic	E.g. suggesting a complete list of scientific literature or blog postings around a topic	None
4. RECOMMEND SEQUENCE	Recommendation of a sequence of items	E.g. receive a proposed sequence of songs	E.g. receiving a proposed sequence through resources to achieve a particular learning goal	Explore formal and informal attributes for representing relevancy to a particular learning goal
5. JUST BROWSING	Recommendations out of the box while user is browsing	E.g. people that bought this, have also bought that	E.g. receiving recommendations for new courses on the university site	Explore formal and informal attributes for representing relevance/usefulness in a learning context
6. FIND CREDIBLE RECOMMENDER	Recommendations during initial exploration/testing phase of a system	E.g. movies that you will definitely like	E.g. restricting course recommendations to ones with high confidence /credibility	Explore criteria for measuring confidence and credibility in formal and informal learning

Source: (MANOUSELIS et al., 2011)

Such need of taking in consideration the learning goal and the learning profile creates a perfect scenario to the application of a strategy for adaptive recommendation. Since it is necessary to look beyond the user preferences to provide a set of suggested resources that will fulfill the recommendation needs. A learner in a TEL domain need to receive, for instance, “a proposed sequence through resources to achieve a particular learning goal”. It is clear in such objective the user expects also such adapted path of resources being interesting and not only related to the fulfillment of the learning goal. The first question to be asked is: which user features (besides her preferences) also need to be taken in consideration when adapting a recommendation in a learning domain?

In (VERBERT et al., 2012) it is shown which profile information a context-aware recommender system in TEL domain should handle to be able to provide a good recommendation. This could be used as a clue to the definition of a user profile to be used in an adaptive recommendation process. The profile defined in the paper should contain the following features:

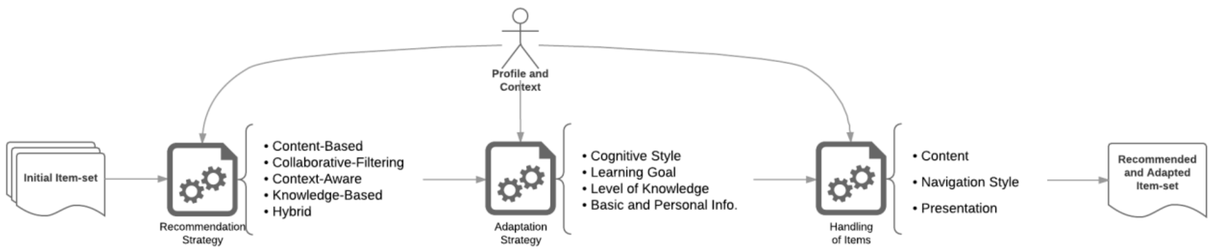
- Computing: information about the computing environment including network, hardware, and software.
- Location: comprises the information about the user geographic position, generally systems use an abstraction to geometric coordinates, such as *classroom*, *home*, *outdoor*.
- Time: is the information about date and time, it is used to stamp a when other contextual information happened.
- Physical Conditions: describes the environmental conditions where the system is in, generally associate information are heat, light, and sound.
- Activity: reflects the tasks, objectives or actions of the user.
- Resource: it is related to information about physical or virtual resources that are target for the recommendation algorithm.
- User: aggregates learner features that have been extensively proposed in the literature of AHS to properly model such users.
    - Basic Personal Information: basic user identification, like *name, identification information, contact information, language capabilities, gender, age, educational level*.
    - Knowledge/Performance: represents user prior knowledge levels.
    - Interests: the most common information used by recommender systems to suggest items. Values can be typically user ratings, terms, tags, comments, and resources read, created, or rated.
    - Learning Goals: distinct between short-term and life-long learning goals.
    - Learning and Cognitive Styles: models the different ways of preferred learning. Some learners can prefer to receive audio, or text, or visual content.
    - Affects: relates to the user emotions during the learning.
    - Background: refers to the user knowledge outside of the core domain of a specific system.
- Social Relations: describe social associations, connections, or affiliations between two or more people.

Between the aforementioned features, the “User” ones are the most important for this thesis scenario. Since it depicts the learner important information to be taken in consideration when providing a recommendation in TEL domain. The other features are most related to the environment where the recommendation is placed.

To illustrate a process of Post-Adaptation the Figure 4.6 shows some examples of recommendation and adaptations strategies possible to be selected to compose the whole adaptive recommendation process. It is possible to have a process with at least one option selected, other options are also available, this figure only shows an example of how an adaptive recommendation process can happen.

The initial item-set can be processed by a content-based, collaborative-filtering, context-aware, knowledge-based, or a hybrid recommendation algorithm. Then such resulting item-set will be also filtered or re-ranked by taking considering the user cognitive style, her learning goal, level of knowledge, background, and other information as shown in (VERBERT et al., 2012). Finally, such set is handled to fulfill the learner specific needs of content, or presentation, or navigation style between the learning resources.

Figure 4.6 - An instance of the post-adaptation process



Source: The author

The process of Adaptation During Recommendation follows a similar path with the difference of the adaptation to user profile features being embedded in the recommendation algorithm, as demonstrated in the previous section.

5 EXPERIMENTS AND RESULTS

In this section it is described an experiment performed to validate the impacts the adaptation to learner profile could cause on a recommendation of educational resources. More specifically, it is used a dataset of movies to be used inside a classroom and it is collected a series of user metrics to check how the adaptation could influence the recommendation.

This chapter is structured as follows, Subsection 5.1 presents the utilized dataset, and how it was collected and organized, 5.2 presents the implemented prototype, and the algorithms that compose it, and 5.3 presents the experiment results.

5.1 Dataset

The dataset used in this research comes from a website named Teach with Movies[1] (TWM), which is a platform where teachers elaborate and made available a series of movie-based lesson plans. The dataset was crawled from the website which provides public access to its content. In Figure 5.1 is shown the website index page, showing a list of highlighted movies lesson plans, a menu categorizing such movies and other links referencing the movies in amazon website.

The website had 426 movies at the time when it was crawled, but from this number only 405 were possible to be used by the recommender system. The reason was because some of

[1] http://www.teachwithmovies.org/

these movies had broken links, some were repeated, some were available by .doc attachments, some had no description, and others were not properly parsed by the crawler.

Figure 5.1 – TWM start page

Source: http://www.teachwithmovies.org/

The crawler was developed in Java by using the Jsoup[2] library to parser the webpages. An additional challenge was to study the pages and to decide which information to parser and how to parser the target information. Since each movie is described by a teacher, and not necessarily the same teacher, we perceived each lesson plan followed a different structure and

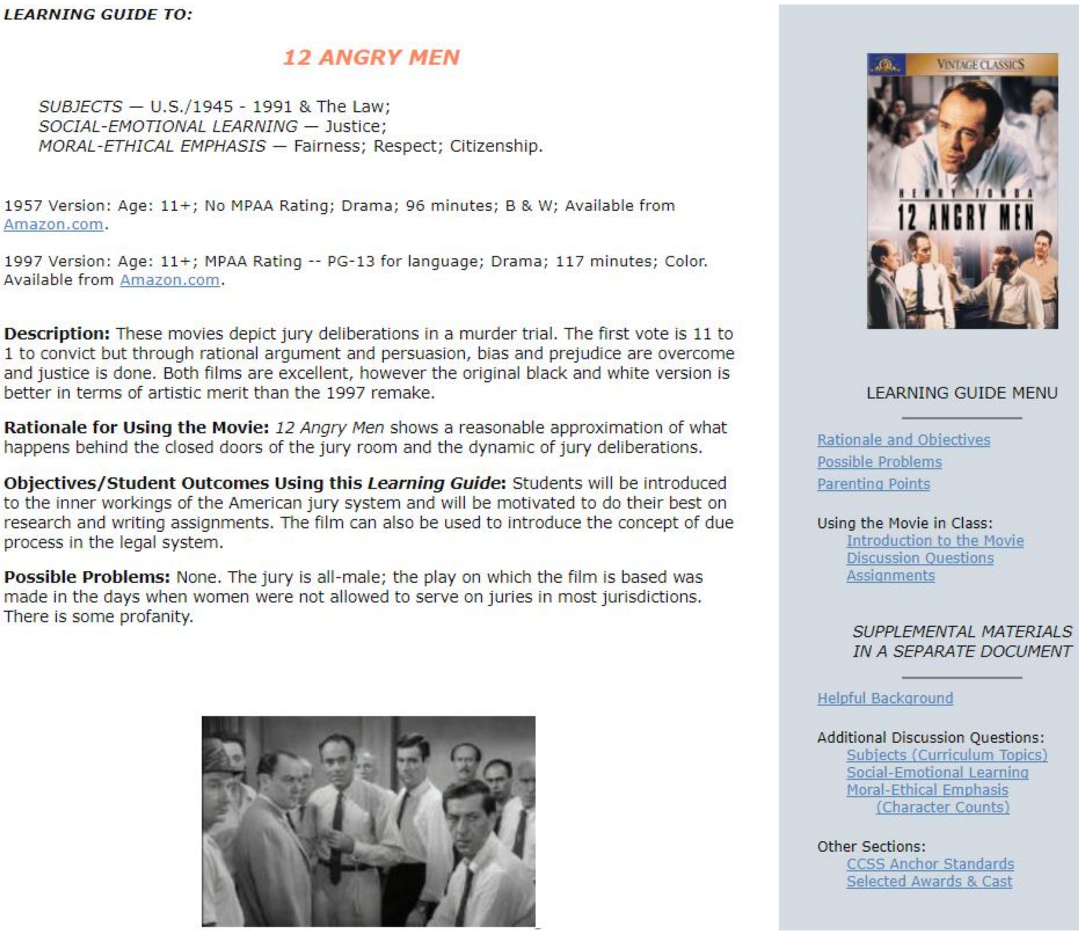
[2] https://jsoup.org/

presented different information about the movies. The only common information about every movie was a recommended age, a title, and some tags of categorization.

Other challenge was to identify what to collect of the lesson plan, since each one presented distinct information. After the manual analysis of a sample of movies it was perceived that most of them presented some "header" information. This information though, was not organized in specific html tags or any other structure to be properly parsed. To this reason it was necessary to combine html description, css tags, and sometimes even the content of a tags to capture the target information.

This "header" consisted of the movie description, the movie rationale, the benefits, the possible problems, and the objectives. However, as shown in Figure 5.2, not all movies presented all these information, in the example it is missing the benefits, but all the other information is presented. Besides such "header" the movies presented other information such questions to use in a class, background to understand the theme, before showing instructions, introduction to the main theme, and so on.

Figure 5.2 – Lesson plan for "12 angry men" movie



LEARNING GUIDE TO:

12 ANGRY MEN

SUBJECTS — U.S./1945 - 1991 & The Law;
SOCIAL-EMOTIONAL LEARNING — Justice;
MORAL-ETHICAL EMPHASIS — Fairness; Respect; Citizenship.

1957 Version: Age: 11+; No MPAA Rating; Drama; 96 minutes; B & W; Available from Amazon.com.

1997 Version: Age: 11+; MPAA Rating -- PG-13 for language; Drama; 117 minutes; Color. Available from Amazon.com.

Description: These movies depict jury deliberations in a murder trial. The first vote is 11 to 1 to convict but through rational argument and persuasion, bias and prejudice are overcome and justice is done. Both films are excellent, however the original black and white version is better in terms of artistic merit than the 1997 remake.

Rationale for Using the Movie: *12 Angry Men* shows a reasonable approximation of what happens behind the closed doors of the jury room and the dynamic of jury deliberations.

Objectives/Student Outcomes Using this *Learning Guide*: Students will be introduced to the inner workings of the American jury system and will be motivated to do their best on research and writing assignments. The film can also be used to introduce the concept of due process in the legal system.

Possible Problems: None. The jury is all-male; the play on which the film is based was made in the days when women were not allowed to serve on juries in most jurisdictions. There is some profanity.

LEARNING GUIDE MENU

Rationale and Objectives
Possible Problems
Parenting Points

Using the Movie in Class:
Introduction to the Movie
Discussion Questions
Assignments

SUPPLEMENTAL MATERIALS IN A SEPARATE DOCUMENT

Helpful Background

Additional Discussion Questions:
Subjects (Curriculum Topics)
Social-Emotional Learning
Moral-Ethical Emphasis (Character Counts)

Other Sections:
CCSS Anchor Standards
Selected Awards & Cast

Source: http://www.teachwithmovies.org/guides/12-angry-men.html

After collecting the website information, it was created a csv file with 8 columns corresponding to: i) lesson plan url; ii) movie description, iii) movie rationale; iv) movie benefits; v) possible problems; vi) objectives; vii) a collection of the “header” without distinction of description, rationale, benefits…; and viii) the movie title. Such file was the information used to populate the database of the prototype described in the next subsection.

5.2 Prototype

To validate the proposition of an adaptive recommendation approach it was built a software prototype for recommendation of the educational movies collected from TWM. Each movie I seen as an educational resource, since the text which characterize it was extracted from the lesson plans presented in TWM website.

The prototype is implemented in Python language using the Django[3] web framework, it was made publicly available through the Heroku Cloud Application Platform[4]. It was implemented two recommendation algorithms, one of matrix factorization as demonstrated in the Adaptation During Recommendation Section, and a traditional TF-IDF Content-Based Filtering. The Factorization Matrix presented the list of recommendations to the user and the Content-Based Filtering was used to recommend similar movies to the user.

The prototype database has 13 tables, but for clarity we present the four main tables in Figure 5.3, the other tables are generated automatically by the Django to facilitate the administration of the database. The central table is the Rating where are stored the user given ratings and the rating predictions. The table Learner stores the educational profile of the user, such profile was built based in the one defined by (VERBERT et al., 2012). The Movie table stores the details of each movie collected from TWM website, more specifically the fields description, rationale, possible problems, objectives, also a movie picture, the movie recommended age, and the movie knowledge area. The User table stores the information about the user account used to access the system, since Django was used as development framework this table was managed automatically by it.

Concerning the relationships, the User table has a one-to-one relationship with the Learner table, meaning each user account is associated with one and only one learner register.

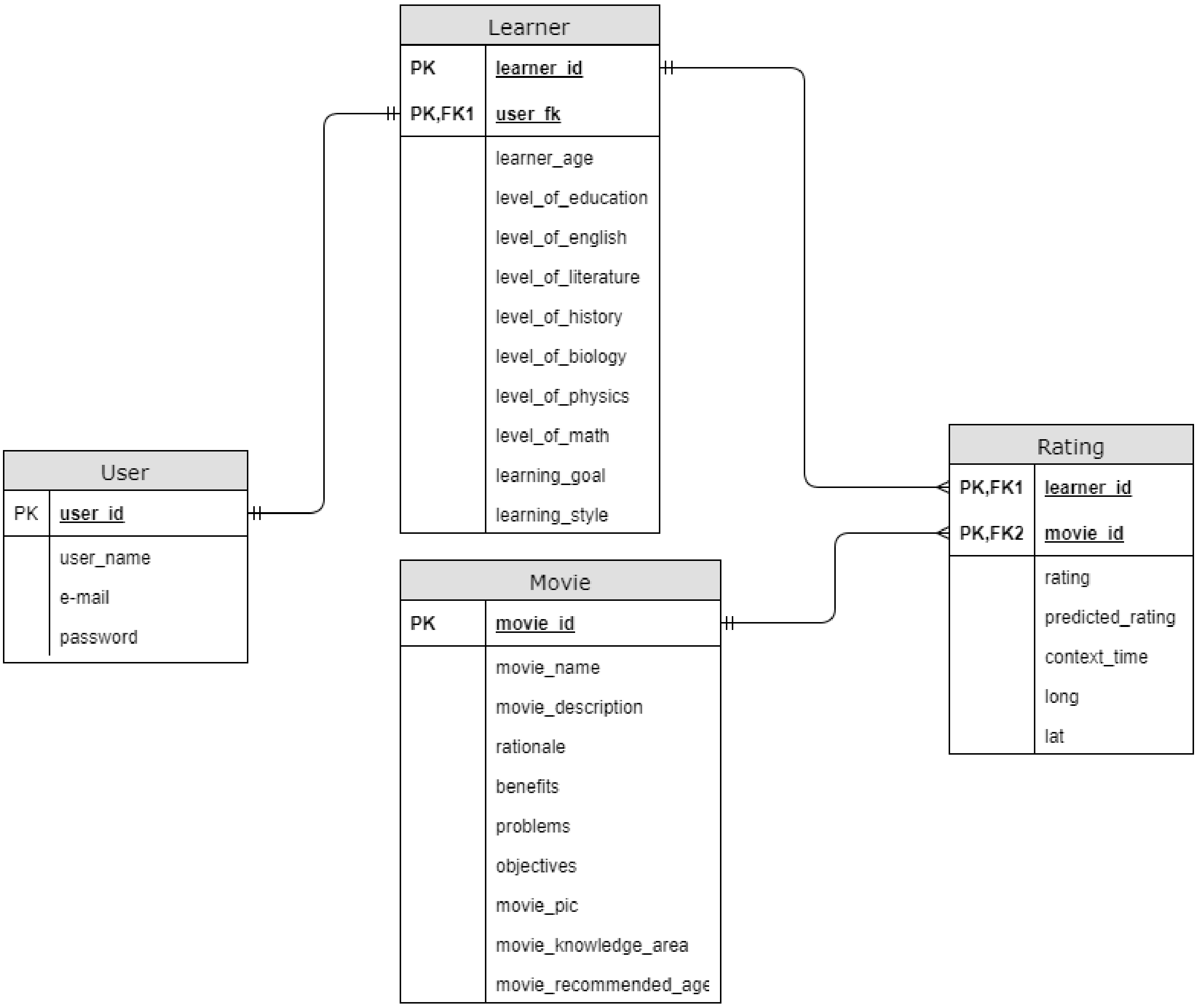
[3] https://www.djangoproject.com/

[4] https://www.heroku.com/

The other two relationships are from Learner and Movie tables to Rating and has the multiplicity of one-to-many, meaning each learner or movie register can have many rating ratings associated with it.

Actually, each learner has *m* rating registers, being *m* = the number of available movies (in this case m=405), and each movie has *n* rating registers, being *n* = the number of users who has rated any movie (in this case n=78). Even if the user has rated a very small portion of the dataset (*sparsity* is very common in recommender systems domains), the system generates a prediction to each movie in the dataset and stores such prediction to later fast access. Such predictions are periodically updated when the server use is low.

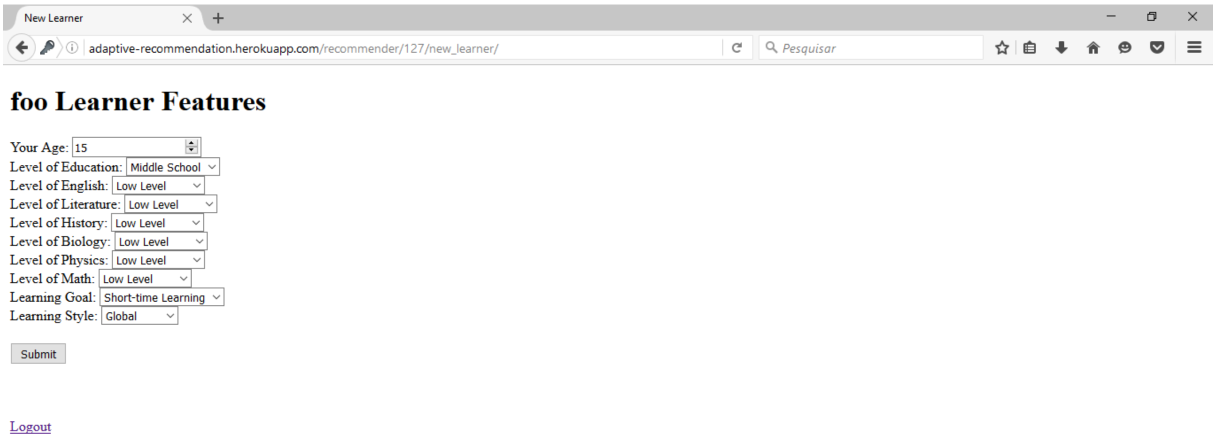
Figure 5.3 – The prototype database schema



Source: The author

The first interaction with the prototype is the login screen, where it is possible to sign in directly or to create a new user in the system. Once the user is signed in for the first time she is asked to create a learner profile, as shown in Figure 5.4.

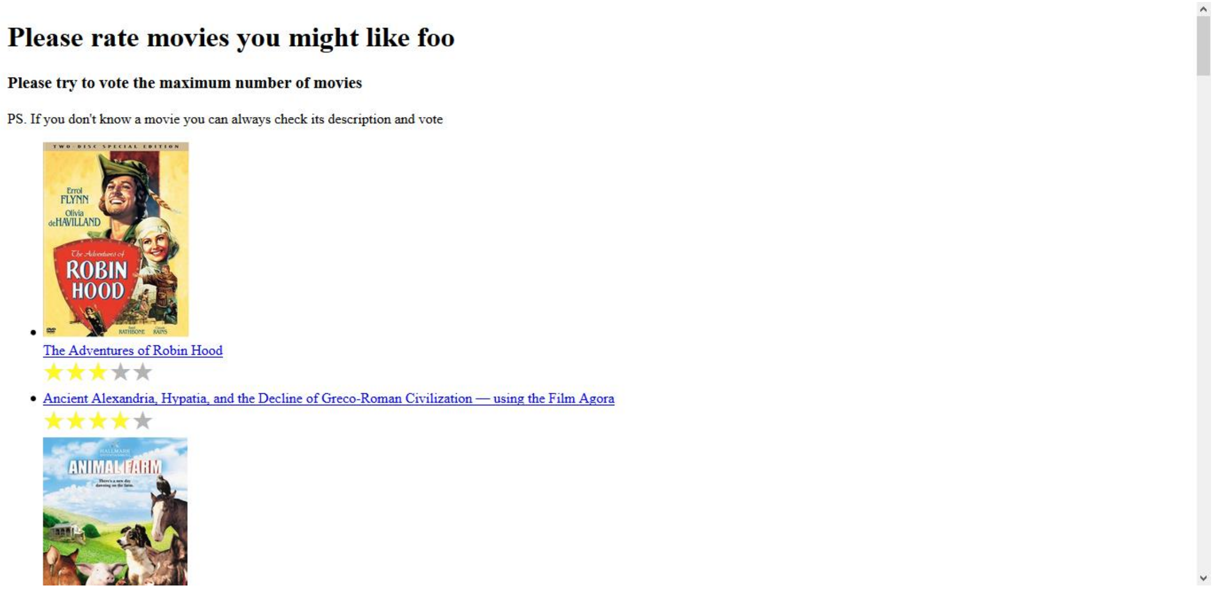
Figure 5.4 – The learner profile screen



Source: The author

It is in this screen where the user will select the different levels of knowledge, her learning goal (Short-Term Learning or Life-long Learning) and her learning style (Sequential or Global). These two learning styles are defined in the classical paper of (FELDER; SOLOMAN, 1991), the authors define some other styles but for this prototype implementation it was selected these two. If the user says she has a more global learning style the system will present the recommendation set as a grid of movies, if she says however, her learning style is sequential the system will present the recommendation set as a sequential list. The other parameters are taken in consideration directly by the recommendation algorithm.

Figure 5.5 – The screen for get user preferences

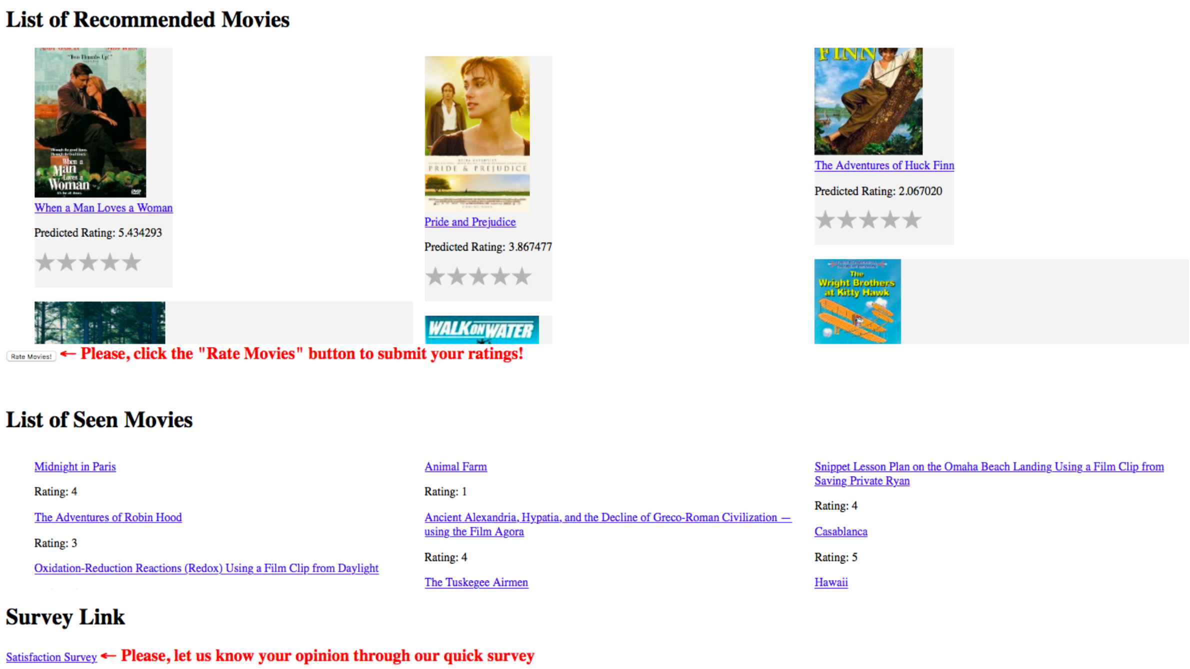


Source: The author

After the learning profile is created, the user provides ratings to a list of randomly selected movies as shown in Figure 5.5. The list brings 20 movies and it is not mandatory the user to rate all the 20 movies, but as more movies she rates more accurate her predicted ratings will be. Some movies did not present a picture in TWM so it was presented only its link in the developed prototype. Other characteristic is sometimes in TWM a movie has its name changed to the subject it is supposed to be used to teach, like “Angels and Demons” is named “Antimatter”, this allows a same movie to appear more than once to support the teaching of different subjects.

In Figure 5.6 is shown the recommendation screen, once the user has filled the learning features and rated the random movies, this will be the main screen she will see every time she logs in. This screen is divided in three parts: i) List of Recommended Movies; ii) List of Seen Movies; and iii) Survey Link. The List of Recommendations presents the movies the user has not rated yet, ordered by descendent value of rating prediction, the user can still give a rating to such movies, which puts them in the second part of the screen. In List of Seen Movies, it is presented a list of the movie the user has already seen (rated), she can still access such movies and see their details, however for space gain purposes it is not shown their pictures in the list. The third and last part of the screen shows a link to the satisfaction survey which is a google form that collects other user data relating the system usage.

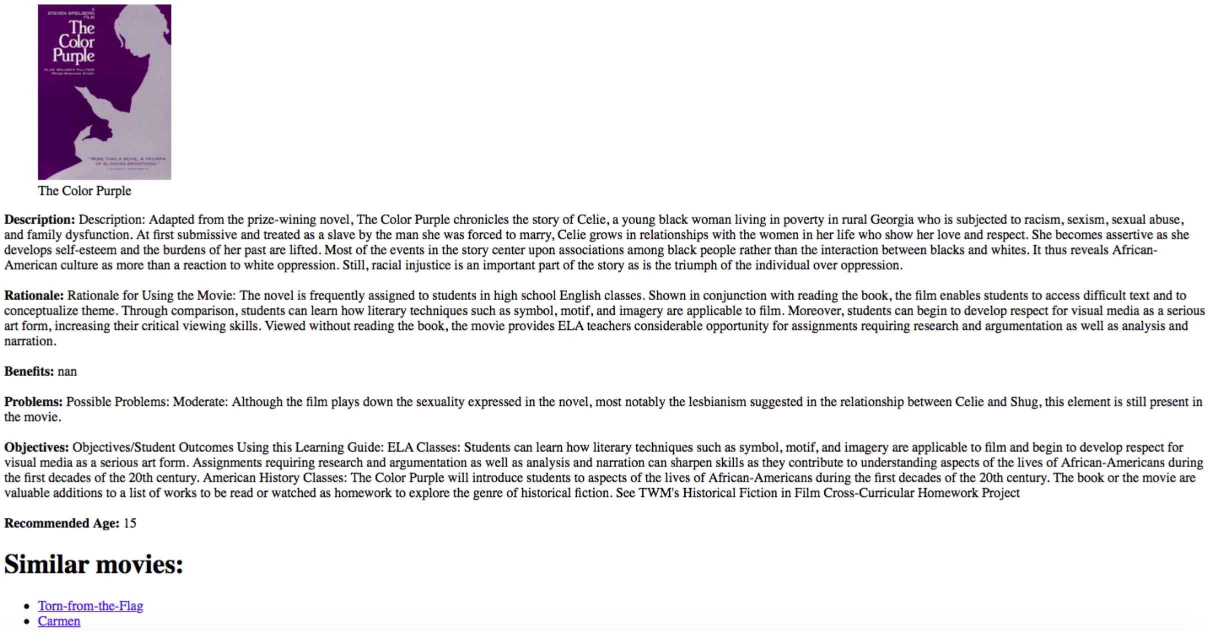
Figure 5.6 – The top-k recommendation screen



Source: The author

If the user wants to access more details about a specific movie she clicks in the link referent to it, Figure 5.7 shows details about the movie “The Color Purple”, differently of a traditional recommender system where it is expected the year of the movie, the cast, the director, and other generic movie information. In this case, the information brought is related as how to use the movie inside a classroom, what are the advantages, the problems, the objectives, how to use it. As already mentioned it is not mandatory to a movie fulfill all the “header” fields, in this case it is also lacking a description of what are the benefits of the movie. However, sometimes the benefits of using it are described in fields like, “rationale”, or “objectives”, like it is in this case. The bottom part of the screen shows a list of other similar movies. This list is got by analyzing the content of the “header” of each movie, it is computed a TF-IDF matrix and then it is made a comparison of what are the most similar movies to it.

Figure 5.7 – The movie details screen



Source: The author

5.3 Experiment Results

To evaluate the developed prototype it was defined a protocol based in the framework presented by (JANNACH; ADOMAVICIUS, 2016). In the paper, the authors define the recommender systems should be evaluated by the consumer and also by the provider's viewpoint. Using the guidelines defined in the framework it was possible to identify the recommendation purposes of the prototype for adaptive recommendation of movies. The purposes were three folded both by the consumer's and by the provider's viewpoint as shown in Table 5.1.

Table 5.1 – The adaptive recommendation purposes

Consumer's viewpoint	**Provider's viewpoint**
1- Show alternatives 2- Help users explore or understand the item space 3- Remember already seen items	a- Change the user behavior in desired directions b- Increase site activity c- Learn about the user.

Source: The author

A successful recommendation should achieve the purposes of both consumer and provider’s viewpoints. To verify such achievements, it is presented in Table 5.2 some related metrics.

Table 5.2 – The metrics related to the recommendation purposes

Consumer’s viewpoint	**Provider’s viewpoint**
1- Click on similar items	a- Clicks on top-k recommendations
2- Satisfaction with items’ presentation	b- Session time
2- Satisfaction with recommended items	c- Prediction accuracy
3- Satisfaction with already seen items	c- User satisfaction with presented profile

Source: The author

Each purpose is verified by one or more metrics, for instance, the purpose of “Show alternatives” are assumed to be verified by the user “Clicks on similar items”. The purpose “Learn about the user” is verified by both the “Prediction accuracy” and by the “User satisfaction with presented profile”.

5.3.1 Data Collect and Statistics

The collection of user clicks was performed by using the Google Analytics tool, both because of its implementation simplicity and for its interface of results that shows the collected data and various visualization forms of it.

The satisfaction metrics were collected through a survey the users have answered after using the prototype, this survey was created through Google Forms and made available in the bottom part of the recommendation screen.

Finally, the accuracy metric was gotten through an offline experiment using the dataset generated during the experiment. A previous training were realized in the algorithms using the Movie Lens dataset (HARPER; KONSTAN, 2015). This training had the goal of defining the best set of parameters to start the recommendations on a new prototype (without any user or preferences registered yet). The user and item bias, demanded by the algorithm of adaptive recommendation, were artificially created to the training phase. Such creation was performed by following the instructions presented in (BALTRUNAS; RICCI, 2009), the biases were then inserted in Movie Lens dataset.

The prototype was made available through internet and its access link was distributed in e-mail lists, in Table 5.3 it is shown some statistics about the system use and the dataset created.

Table 5.3 – Dataset Statistics

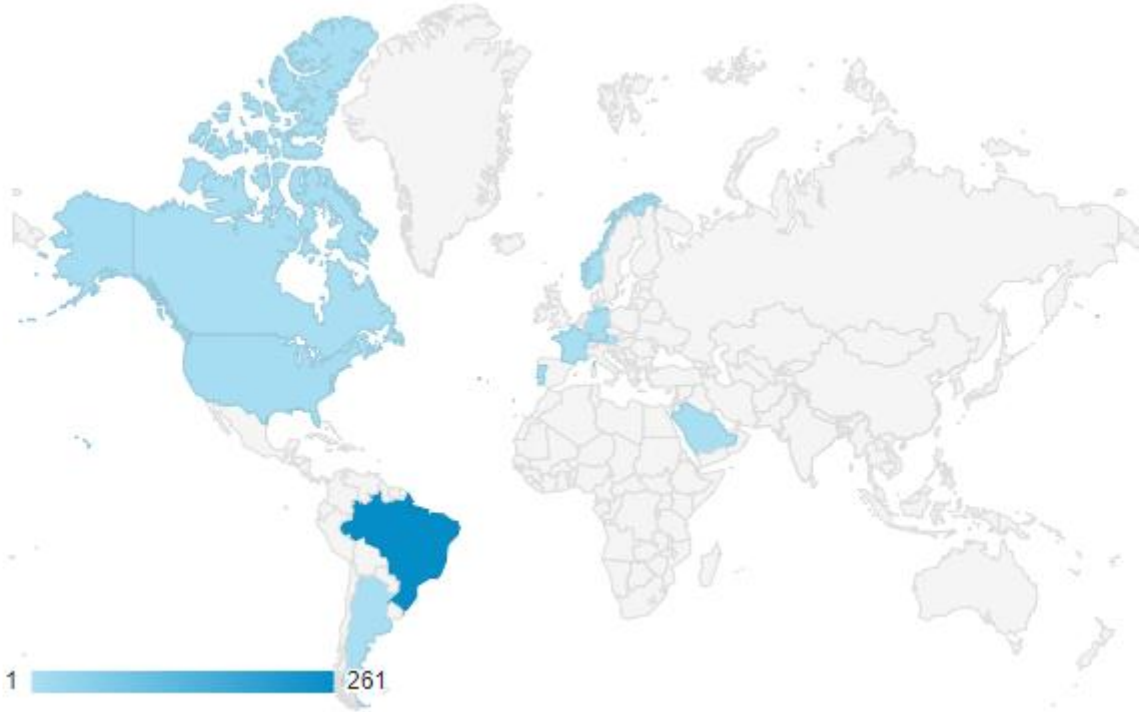
Variable analyzed	**Value**
Number of unique system accesses	*262*
Number of registered system accounts	*126*
Number of registered learner profiles	*82*
Number of learners who rated some movie	*78*
Number of explicit ratings	*2426*
Number of movies	*405*

Source: The author

The numbers in the table above shows that 48% of the people who accessed the system has registered a user account, 65% of the registered account also have filled the learner profile, an 95% of the registered learner have also rated some movie. Since 78 was the number of users who have effectively utilized the system, from now on whenever we refer to system users means these 78 registered learners who also have rated some movie. Each user has rated 31 movies on average, this number of ratings provided a good basis to build the user's profile. The dataset analysis also has shown that all the 405 movies have been rated at least once.

The users had public access to the system during one moth and since it was available through internet in English language, the users' demographic was as diverse as possible, as shown in Figure 5.8. The countries in blue have users who accessed the system, as darkest the blue is means the country had more sessions. Brazil was the top country presenting 261 sessions in the system, followed by United States of America which had 6 sessions, and Portugal with 3 sessions.

Figure 5.8 - Accesses by country

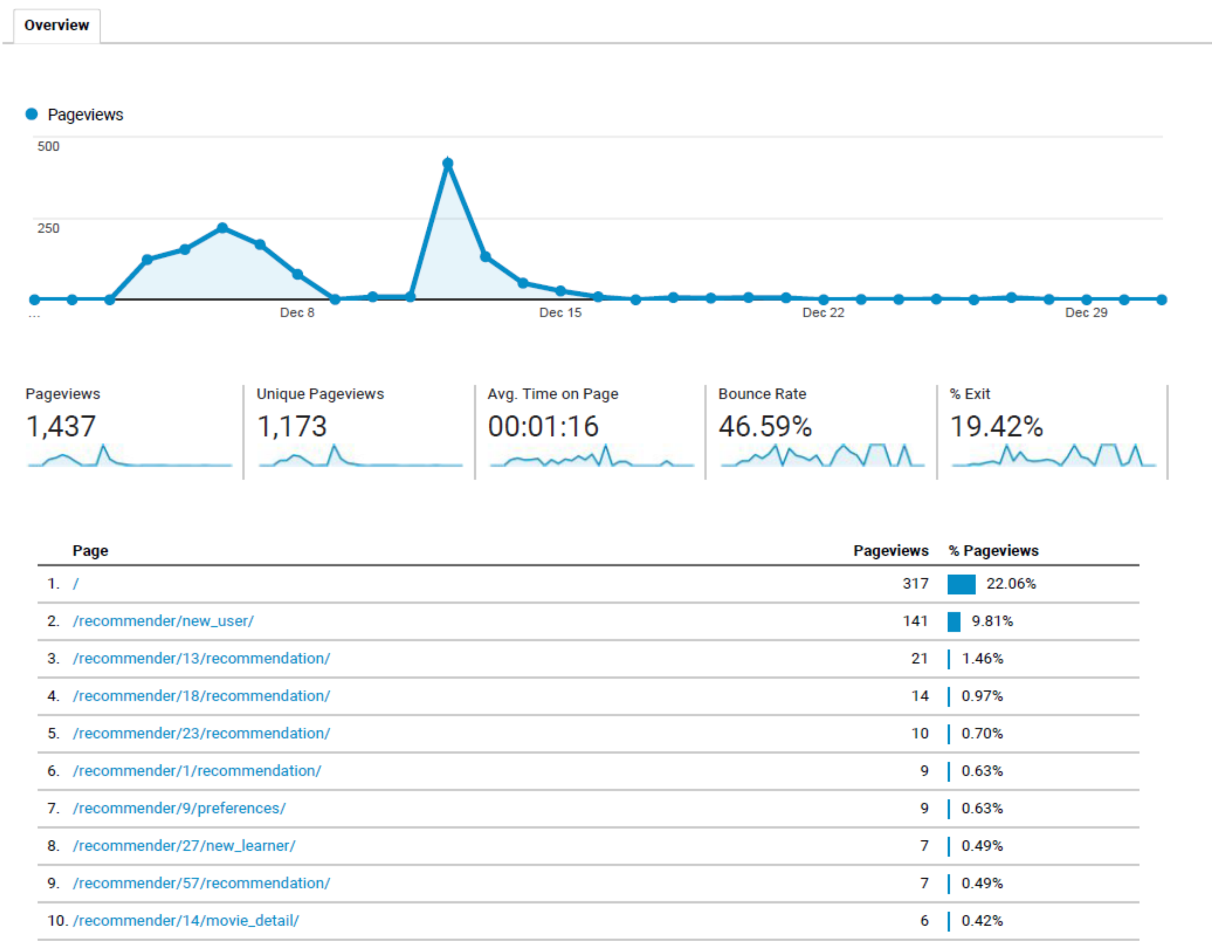


Source: The author

From the Google Analytics data it was possible to already verify three defined metrics. Relating the clicks in similar items and the clicks in top-k recommendations, the report analysis has shown that 69,38% of the clicks the users gave in the system were in movies. From all movies clicks 90,43% were given in top-k recommendations and only 9,57% in recommendation of similar movies. We believe one explanation for such difference is how the system interface is built, since the main screen of the system is the screen of recommendation, to have access to the list of similar movies the user must open one of the movie details screen and then click on a similar movie. However, having almost 70% of all the system clicks on movie recommendations gives a good clue about the system influence on user behavior. It is possible affirm the system is better in conducting user behavior to the recommendations than presents a list of alternatives.

Other metric verified through Google Analytics data is the session time, which tries to verify if the system has increase site activity. According to a study presented by (LIU; WHITE; DUMAIS, 2010), 80% of webpages can maintain their user for no more than 70 seconds. Looking to our statistics collected through Google Analytics (Figure 5.9), the system has an average time on page of 76 seconds, being put between the 20% of websites which can grab the user for more than 70 seconds, giving a clue that the approach can grab the user attention for more than the average time spent in a web page.

Figure 5.9 - The system accesses overview



	Page	Pageviews	% Pageviews
1.	/	317	22.06%
2.	/recommender/new_user/	141	9.81%
3.	/recommender/13/recommendation/	21	1.46%
4.	/recommender/18/recommendation/	14	0.97%
5.	/recommender/23/recommendation/	10	0.70%
6.	/recommender/1/recommendation/	9	0.63%
7.	/recommender/9/preferences/	9	0.63%
8.	/recommender/27/new_learner/	7	0.49%
9.	/recommender/57/recommendation/	7	0.49%
10.	/recommender/14/movie_detail/	6	0.42%

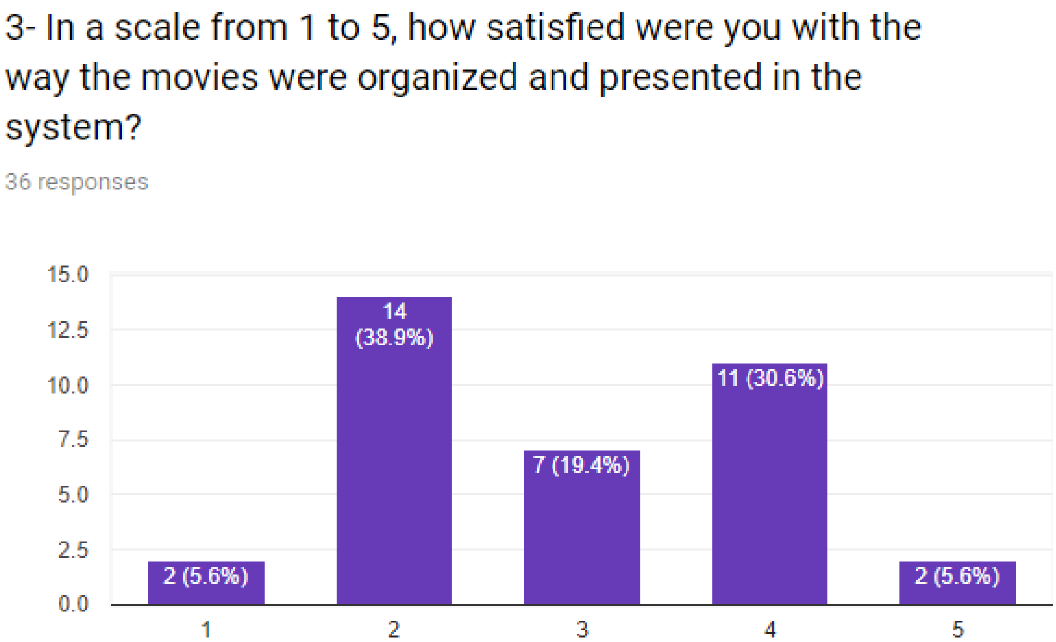
Source: The author

The Figure 5.9 also shows data about when the system had more accesses, the most view pages, and the bounce rate. Looking to the list of most view pages it is also confirmed the most accessed page was the recommendation screen instead of the screen of movie details. This confirms the results given by the analysis of click data; the system is better in conducting the user behavior to the top-k adaptive recommendations than conducting her in the navigation between the list of similar movies.

The satisfaction metrics collected through the Google Form, were answered by 36 users from 15 different research groups, being the most popular the PPGC-UFRGS. Other result is concerning to the user profile satisfaction, most of the users (15 of them specifically) found it “a little confuse”. The reason for this was better explained in the section of suggestions, most of the users found difficult to select a learning style and a learning goal, because they were unfamiliar with the options. Despite the tooltips presented explaining each option, the users still had difficult in distinguishing the differences between the learning style sequential and global. However, 14 users answered the profile structure was “good enough”, this result leads the conclusion that we cannot affirm the users were satisfied with the learner profile nor we can affirm they were very dissatisfied.

The results about the satisfaction with the items presentation is shown in Figure 5.10. This was the worst result collected in the survey, most of the users complained about the system interface being kept so simple and it brought difficult in understanding and exploring the item space.

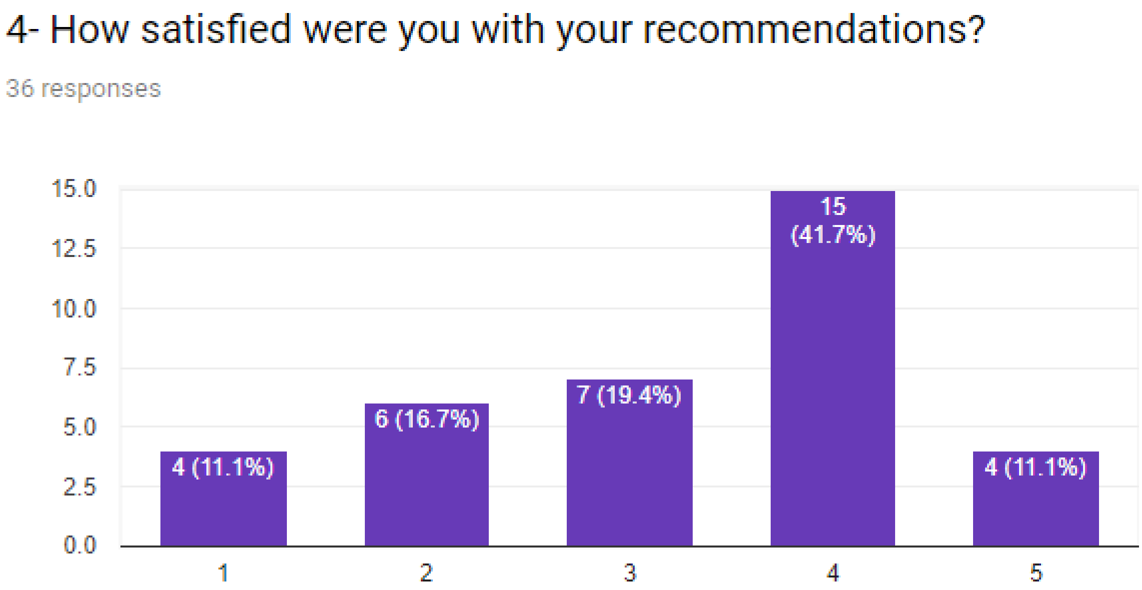
Figure 5.10 - Items presentation Satisfaction



Source: The author

About the satisfaction with the recommendations, most of the users found the recommendations very good, giving a 4 in a 1 to 5 scale of evaluation, as shown in Figure 5.11.

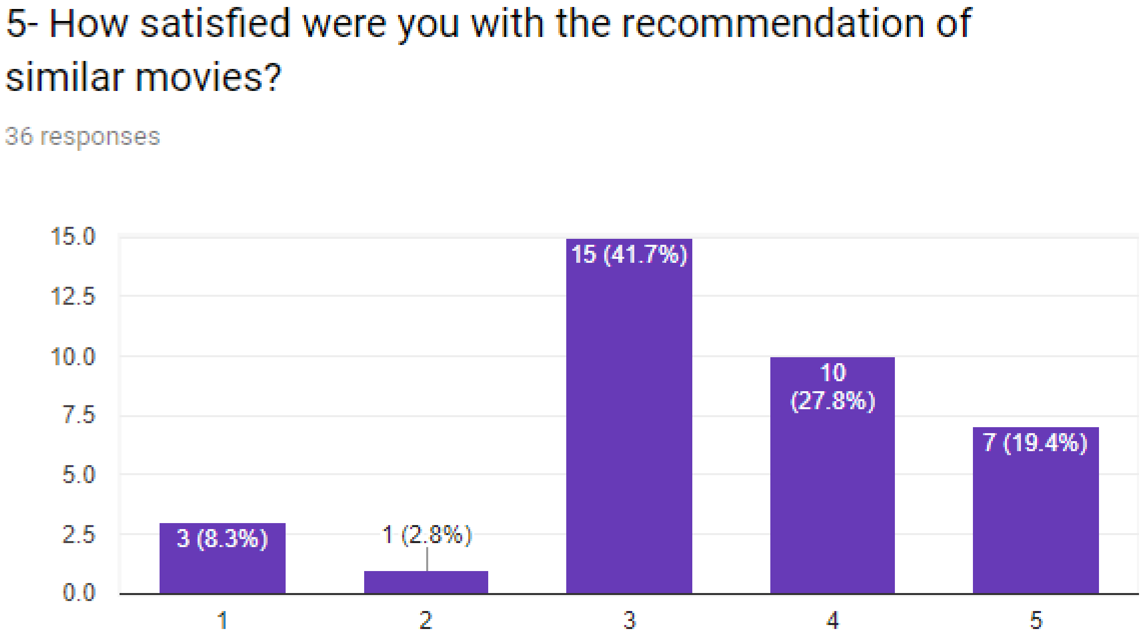
Figure 5.11 - Recommendations Satisfaction



Source: The author

The satisfaction with similar movies recommendation, were considered neutral, since most of the users gave a 3 to evaluate this type of recommendation, the result is shown in Figure 5.12. One of the reasons that can be responsible for such result is the interface choice of putting such list inside the screen of movie details, because of this most of the user barely has navigated through the list of similar movies. We believe a different choice of interface, highlighting this type of recommendation, can stimulate the use of it and help the users to develop a stronger opinion over such type of recommendation.

Figure 5.12 - Satisfaction of similar movies

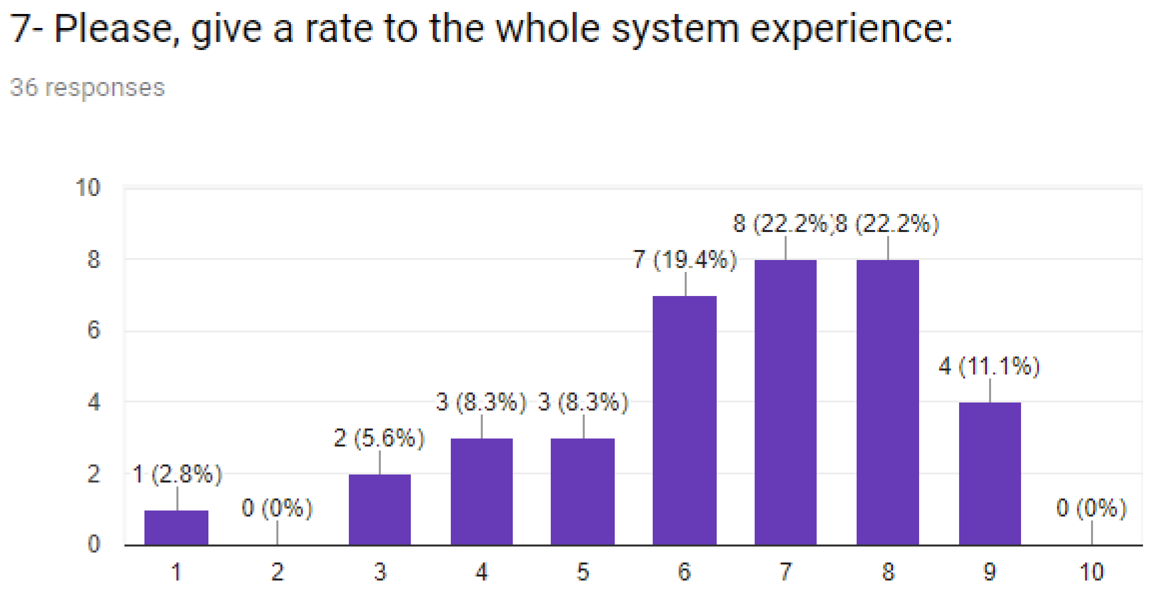


Source: The author

The satisfaction with the list of already seen movies has shown that 77,8% of the users did not find interesting have this information available. This was an unexpected result, since it is one of the user's purpose of using the system "Remember already seen items". However, our user did not find interesting to have a list of such items available.

Regarding the user satisfaction with the whole system experience, it was provided a scale from 1 to 10, and as shown in Figure 5.13 almost 75% of the users has given a rating between 6 to 9. This is a motivating result, since it tries to capture the user general feeling about using the system, and despite some complaints mainly about the simple interface most of the users had a positive experience by using the system. We believe the quality of the recommendations weighted more than a simple interface in this decision.

Figure 5.13 - User overall satisfaction



Source: The author

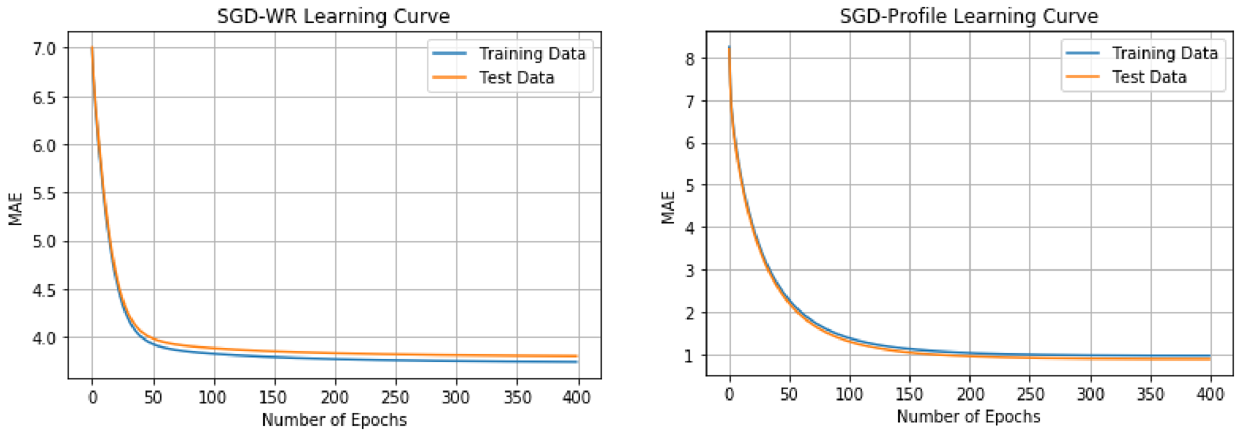
Finally, the last metrics is related to the system accuracy to predict the user rating. This metric was verified through an offline experiment where the algorithm for adaptive recommendation was compared with a traditional matrix-factorization algorithm. Both were implemented in python and compared using the Mean Absolute Error (MAE), which is a statistic metric that measures the difference between two continuous variables.

A series of parameters were also experimented and the ones which returned better results were:

$$\lambda = 0.9 \text{ (Regularization weight)}$$
$$k = 8 \text{ (Number of latent factors)}$$
$$number\ of\ epochs = 400$$
$$\gamma = 0.009 \text{ (learning rate)}$$

The accuracy results are shown using a learning curve in Figure 5.14 (a) and Figure 5.14 (b) being the figure b related to the adaptive recommendation algorithm. The curves presented some differences between the approaches, while a traditional matrix-factorization algorithm reaches a plateau of learning near the 50 epochs, the algorithm for adaptive recommendation reaches such plateau near the 150 epochs. However, while the traditional matrix-factorization presents an error of 3.79 after 400 epochs, the adaptive recommendation algorithm achieves an error of 0.88 after the same number of epochs. This result proves that the algorithm for adaptive recommendation is considerably more accurate than a traditional matrix-factorization in predicting ratings. It is also noticeable the different scales between the two figures, while figure (a) starts with an error of 7 and descends to 3.79, figure (b) starts with a similar error 8.3 approximately and descends up to 0.88.

Figure 5.14 - Learning curve for accuracy in rating prediction



(a) (b)

Source: The author

In a nutshell, the results presented above have shown the prototype can generate good recommendations but need to have an improvement in its interface. Despite of that the users were satisfied using the system and its recommendations. The quality of rating prediction was also considerably more accurate when compared to a traditional matrix-factorization algorithm.

6 CONCLUSIONS AND FUTURE WORKS

Recommenders systems became a widespread tool to suggest interesting items to a variety of users. From its first proposition to now, such systems have assumed different kinds of roles, and to fulfill the needs of such roles it is necessary to provide more than good recommendations. The recommendation process need to better understand the user and her demands to provide a list of items that is as personalized as possible to such user. One of the suggestions found in the literature is to look to adaptive hypermedia systems as a starting point, since such systems were firstly conceived to solve the problem of *one-size-fits-all*, where the same resource is given to different kinds of users.

The problem of adapting to respond to new user demands is also very related to ubiquitous environments, where the constant change of user status (location, movements, velocity) and also goals push the systems into the need of adapting their behavior to follow the user dynamicity. In this scenario, it was performed a systematic mapping of the literature to find ubiquitous recommender approaches that also provided an adaptation feature. The conclusions of this mapping have shown firstly the almost inexistence of recommender approaches that also embed an adaptation feature, and secondly the need of adaptation goes beyond the ubiquitous domain and it is a problem intrinsic of the whole recommendation process.

In this context, this thesis has discussed the importance an adaptation feature should have to a recommendation process. It is also structured an approach to provide adaptation in recommender systems, where it is commented the advantages and drawbacks in putting the adaptation before, after, or during the recommendation algorithm.

The experiments to validate such approach were performed using the new algorithm for adaptation during the recommendation. Such algorithm takes in consideration a number of user profile features and take such features to the computing of the predicted rating. Each profile feature contributes with a bias to the estimation of the user rating. For instance, if the user has a low level of English, and it is known the users with low level of English give a rating that is -0.4 points below an item average rating; the rating estimator will consider such profile bias.

During the experiments, such bias consideration has shown to influence positively in the rating prediction, and also with the user satisfaction regarding the recommendations. Our results have shown the novel algorithm demands more time to learn but it is considerably more accurate than a traditional matrix-factorization one. One of the reasons for such accuracy difference is the consequence of the deeper user knowledge available to the algorithm, so it is able to provide good recommendations even with a smaller dataset. This is an important conclusion since the algorithm has shown a better ability to handle *cold-start* situations when compared to the traditional matrix-factorization one. The profile biases insert knowledge about the user directly in the rating prediction, making the algorithm able to deal better with smaller datasets.

Besides accuracy, the approach also has shown very good results in grabbing user attention, since it was able to grab the user attention for more than 70 seconds. Concerning user satisfaction, the prototype has presented very good results, the critical point in this evaluation was the system simple interface.

Considering such results, we expect in the future, to enhance the system interface to provide a better user experience and consequently improve the user satisfaction. It is also expected to develop a dataset with other types of resources, in other language than English to compare the effectiveness of the approach in recommending to different domains using different kinds of information.

6.1 List of Publications

During the Ph.D. some papers related to this thesis were publish, they are:

[1] J. K. Kambara, G. M. Machado, L. H. Thom, and L. K. Wives, “Business Process Modeling and Instantiation in Home Care Environments,” in *International Conference on Enterprise Information Systems*, 2014.

[2] G. M. Machado and J. P. M. de Oliveira, “Context-aware adaptive recommendation of resources for mobile users in a university campus,” in *2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2014, pp. 427–433.

[3] G. M. Machado, J. Palazzo, and M. De Oliveira, “CARLO : Modelo Ontológico de Contexto para Recomendação de Objetos de Aprendizagem em Ambientes Pervasivos,” in *Anais do Simpósio Brasileiro de Bancos de Dados*, 2014, pp. 47–56.

[4] G. M. Machado, V. Maran, I. Gasparini, A. M. Pernas, and J. P. M. de Oliveira, “Uma Revisão Sistemática sobre as Abordagens Ubíquas para Recomendação Educacional : Estariam Elas se Tornando Adaptativas ?,” in *Anais do Simpósio Brasileiro de Informática na Educação*, 2015.

[5] V. Maran, G. M. Machado, G. A. Degradai, J. P. M. de Oliveira, and S. Maldaner, “Revisão

Sistemática de Modelos Conceituais e Ontologias para Representação de Dados Micrometeorológicos," in *IX Workshop Brasileiro de Micrometeorologia*, 2015.

[6] G. M. Machado, V. Maran, L. P. Dornelles, I. Gasparini, L. H. Thom, and J. P. M. de Oliveira, "A systematic mapping on adaptive recommender approaches for ubiquitous environments," *Computing*, vol. 99, pp. 1–27, Aug. 2017.

[7] V. Woloszyn, G. M. Machado, J. Palazzo Moreira de Oliveira, L. Krug Wives, and H. Saggion, "BEATnIk: an algorithm to automatic generation of educational description of movies," in *Proceedings of the SBIE 2017*, 2017, pp. 1377–1386.

[8] V. Maran, G. M. Machado, A. Machado, I. Augustin, J. C. D. Lima, and J. P. M. de Oliveira, "Database Ontology-Supported Query for Ubiquitous Environments," in *Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web*, 2017, pp. 185–188.

[9] V. Maran, G. Medeiros Machado, A. Machado, I. Augustin, and J. Palazzo M. de Oliveira, "UPCaD: A Methodology of Integration Between Ontology-Based Context-Awareness Modeling and Relational Domain Data," *Information* , vol. 9, no. 2. 2018.

REFERENCES

ABECH, M. et al. A Model for Learning Objects Adaptation in Light of Mobile and Context-aware Computing. **Personal Ubiquitous Comput.**, v. 20, n. 2, p. 167–184, abr. 2016.

ADOMAVICIUS, G. et al. Incorporating contextual information in recommender systems using a multidimensional approach. **ACM Transactions on Information Systems**, v. 23, n. 1, p. 103–145, 1 jan. 2005.

ADOMAVICIUS, G.; TUZHILIN, A. Context-Aware Recommender Systems. In: RICCI, F. et al. (Eds.). . **Recommender Systems Handbook SE - 7**. [s.l.] Springer US, 2011. p. 217–253.

ADOMAVICIUS, G.; TUZHILIN, A. Context-Aware Recommender Systems. In: RICCI, F.; ROKACH, L.; SHAPIRA, B. (Eds.). . **Recommender Systems Handbook**. Boston, MA: Springer US, 2015. p. 191–226.

ANACLETO, R. et al. Mobile application to provide personalized sightseeing tours. **Journal of Network and Computer Applications**, v. 41, n. 0, p. 56–64, maio 2014.

ARNABOLDI, V. et al. A personalized recommender system for pervasive social networks. **Pervasive and Mobile Computing**, ago. 2016.

ATIF, Y.; MATHEW, S.; LAKAS, A. Building a smart campus to support ubiquitous learning. **Journal of Ambient Intelligence and Humanized Computing**, p. 1–16, 2014.

ATIF, Y.; MATHEW, S. S.; LAKAS, A. Building a smart campus to support ubiquitous learning. **Journal of Ambient Intelligence and Humanized Computing**, v. 6, n. 2, p. 223–238, 2015.

BAGCI, H.; KARAGOZ, P. **Random walk based context-aware activity recommendation for location based social networks**. 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA). **Anais**...IEEE, out. 2015

BALTRUNAS, L.; LUDWIG, B.; RICCI, F. **Matrix factorization techniques for context aware recommendation**. Proceedings of the fifth ACM conference on Recommender systems. **Anais**...: RecSys ’11.New York, NY, USA: ACM, 2011

BALTRUNAS, L.; RICCI, F. **Context-based splitting of item ratings in collaborative**

filtering. Proceedings of the third ACM conference on Recommender systems - RecSys '09. **Anais**...: RecSys '09.New York, New York, USA: ACM Press, 2009

BEAMON, B.; KUMAR, M. **Adaptive Context Reasoning in Pervasive Systems**. Proceedings of the 9th International Workshop on Adaptive and Reflective Middleware. **Anais**...: ARM '10.New York, NY, USA: ACM, 2010

BEER, W.; DERWEIN, C.; HERRAMHOF, S. **Implementation of Context-Aware Item Recommendation through MapReduce Data Aggregation**. Proceedings of International Conference on Advances in Mobile Computing & Multimedia - MoMM '13. **Anais**...: MoMM '13.New York, New York, USA: ACM Press, 2013Disponível em: <http://doi.acm.org/10.1145/2536853.2536859>. Acesso em: 15 ago. 2014

BELLAVISTA, P. et al. A survey of context data distribution for mobile ubiquitous systems. **ACM Computing Surveys**, v. 44, n. 4, p. 1–45, 1 ago. 2012.

BENOUARET, I.; LENNE, D. **Personalizing the Museum Experience through Context-Aware Recommendations**. 2015 IEEE International Conference on Systems, Man, and Cybernetics. **Anais**...IEEE, out. 2015

BIANCALANA, C. et al. An Approach to Social Recommendation for Context-aware Mobile Services. **ACM Trans. Intell. Syst. Technol.**, v. 4, n. 1, p. 10:1--10:31, 2013.

BÖHMER, M.; BAUER, G. **Exploiting the Icon Arrangement on Mobile Devices As Information Source for Context-awareness**. Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services. **Anais**...: MobileHCI '10.New York, NY, USA: ACM, 2010

BOURKE, S.; MCCARTHY, K.; SMYTH, B. **The Social Camera: A Case-study in Contextual Image Recommendation**. Proceedings of the 16th International Conference on Intelligent User Interfaces. **Anais**...: IUI '11.New York, NY, USA: ACM, 2011

BRUSILOVSKY, P. Adaptive Hypermedia. **User Modeling and User-Adapted Interaction**, v. 11, n. 1–2, p. 87–110, 2001.

BRUSILOVSKY, P. et al. Open Social Student Modeling for Personalized Learning. **IEEE Transactions on Emerging Topics in Computing**, v. 4, n. 3, p. 450–461, jul. 2016.

BRUSILOVSKY, P.; MILLÁN, E. User Models for Adaptive Hypermedia and Adaptive Educational Systems. In: BRUSILOVSKY, P.; KOBSA, A.; NEJDL, W. (Eds.). . **The Adaptive Web Methods and Strategies of Web Personalization**. [s.l.] Springer Berlin Heidelberg, 2007. p. 3–53.

BUCHANAN, G. R. **The Fused Library: Integrating Digital and Physical Libraries with Location-aware Sensors**. Proceedings of the 10th Annual Joint Conference on Digital Libraries. **Anais**...: JCDL '10.New York, NY, USA: ACM, 2010

CALERO VALDEZ, A.; ZIEFLE, M.; VERBERT, K. **HCI for Recommender Systems: The Past, the Present and the Future**. Proceedings of the 10th ACM Conference on Recommender Systems. **Anais**...: RecSys '16.New York, NY, USA: ACM, 2016

CHENG, Z.; SHEN, J. **Just-for-Me: An Adaptive Personalization System for Location-Aware Social Music Recommendation**. Proceedings of International Conference on Multimedia Retrieval. **Anais**...: ICMR '14.New York, NY, USA: ACM, 2014

CHIN, A. et al. **Linking People Through Physical Proximity in a Conference**. Proceedings of the 3rd International Workshop on Modeling Social Media. **Anais**...: MSM '12.New York, NY, USA: ACM, 2012

CHIN, A. et al. Connecting People Through Physical Proximity and Physical Resources at a Conference. **ACM Trans. Intell. Syst. Technol.**, v. 4, n. 3, p. 50:1--50:21, 2013.

COLOMO-PALACIOS, R. et al. Towards a social and context-aware mobile recommendation system for tourism. **Pervasive and Mobile Computing**, abr. 2016.

CONSOLE, L. et al. Interacting with Social Networks of Intelligent Things and People in the World of Gastronomy. **ACM Trans. Interact. Intell. Syst.**, v. 3, n. 1, p. 4:1--4:38, 2013.

DE BRA, P. Adaptive Hypermedia. In: ADELSBERGER, H. et al. (Eds.). . **Handbook on Information Technologies for Education and Training SE - 2**. International Handbooks on Information Systems. [s.l.] Springer Berlin Heidelberg, 2008. p. 29–46.

DE GEMMIS, M. et al. Semantics-Aware Content-Based Recommender Systems. In: RICCI, F.; ROKACH, L.; SHAPIRA, B. (Eds.). . **Recommender Systems Handbook**. Boston, MA: Springer US, 2015. p. 119–159.

DORYAB, A.; BARDRAM, J. E. **Designing Activity-aware Recommender Systems for Operating Rooms**. Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation. **Anais**...: CaRR '11.New York, NY, USA: ACM, 2011

EKSTRAND, M. D. et al. **Letting Users Choose Recommender Algorithms: An Experimental Study**. Proceedings of the 9th ACM Conference on Recommender Systems. **Anais**...: RecSys '15.New York, NY, USA: ACM, 2015

ELHAMDAOUI, A.; ABIK, M.; AJHOUN, R. Platform for Dynamic Adaptation of Learning Content for Pervasive Learning Environments. In: SNASEL, V.; PLATOS, J.; EL-QAWASMEH, E. (Eds.). . **Digital Information Processing and Communications SE - 45**. Communications in Computer and Information Science. [s.l.] Springer Berlin Heidelberg, 2011. v. 189p. 518–529.

EVERS, C. et al. The user in the loop: Enabling user participation for self-adaptive applications. **Future Generation Computer Systems**, v. 34, n. 0, p. 110–123, maio 2014.

FAN, X. et al. **Context-Aware Ubiquitous Web Services Recommendation Based on User Location Update**. 2015 International Conference on Cloud Computing and Big Data (CCBD). **Anais**...IEEE, nov. 2015

FELDER, R. M.; SOLOMAN, B. A. **Index of learning styles**, 1991.

FRANCO, L. K. et al. MUCS: A model for ubiquitous commerce support. **Electronic Commerce Research and Applications**, v. 10, n. 2, p. 237–246, mar. 2011.

GASPARINI, I. et al. **Combining Ontologies and Scenarios for Context-aware e-Learning Environments**. Proceedings of the 28th ACM International Conference on Design of Communication. **Anais**...: SIGDOC '10.New York, NY, USA: ACM, 2010

GOLDBERG, D. et al. Using collaborative filtering to weave an information tapestry. **Communications of the ACM**, v. 35, n. 12, p. 61–70, 1 dez. 1992.

GUO, B. et al. **MobiGroup: Enabling Lifecycle Support to Social Activity Organization and Suggestion With Mobile Crowd SensingIEEE Transactions on Human-Machine Systems**, 2016.

HALLER, M. **Implementing your own Recommender Systems in Python using Stochastic**

Gradient Descent. Disponível em: <http://online-dev.cambridgecoding.com/notebooks/mhaller/implementing-your-own-recommender-systems-in-python-using-stochastic-gradient-descent-4>. Acesso em: 21 jan. 2017.

HARPER, F. M. et al. **Putting Users in Control of Their Recommendations**. Proceedings of the 9th ACM Conference on Recommender Systems. **Anais**...: RecSys ’15.New York, NY, USA: ACM, 2015

HARPER, F. M.; KONSTAN, J. A. The MovieLens Datasets: History and Context. **ACM Trans. Interact. Intell. Syst.**, v. 5, n. 4, p. 19:1--19:19, dez. 2015.

HERLOCKER, J.; KONSTAN, J. A.; RIEDL, J. An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms. **Information Retrieval**, v. 5, n. 4, p. 287–310, 2002.

HERLOCKER, J. L. et al. Evaluating collaborative filtering recommender systems. **ACM Transactions on Information Systems**, v. 22, n. 1, p. 5–53, 1 jan. 2004.

HONOLD, F.; SCHÜSSEL, F.; WEBER, M. **Adaptive Probabilistic Fission for Multimodal Systems**. Proceedings of the 24th Australian Computer-Human Interaction Conference. **Anais**...: OzCHI ’12.New York, NY, USA: ACM, 2012

HSU, C.-C.; HO, C.-C. The design and implementation of a competency-based intelligent mobile learning system. **Expert Systems with Applications**, v. 39, n. 9, p. 8030–8043, jul. 2012.

IEEE Standard for Learning Object Metadata. **IEEE Std 1484.12.1-2002**, 2002.

JANNACH, D. et al. **Recommender Systems**. Cambridge: Cambridge University Press, 2010.

JANNACH, D.; ADOMAVICIUS, G. **Recommendations with a Purpose**. Proceedings of the 10th ACM Conference on Recommender Systems. **Anais**...: RecSys ’16.New York, NY, USA: ACM, 2016

KAMBARA, J. K. et al. **Business Process Modeling and Instantiation in Home Care Environments**. International Conference on Enterprise Information Systems. **Anais**...2014

KAPOOR, K. et al. **“I Like to Explore Sometimes”: Adapting to Dynamic User Novelty**

Preferences. Proceedings of the 9th ACM Conference on Recommender Systems. **Anais**...: RecSys ’15.New York, NY, USA: ACM, 2015

KAPTEIN, M.; VAN HALTEREN, A. Adaptive persuasive messaging to increase service retention: using persuasion profiles to increase the effectiveness of email reminders. **Personal and Ubiquitous Computing**, v. 17, n. 6, p. 1173–1185, 11 jul. 2012.

KEELE, S. **Guidelines for performing systematic literature reviews in software engineering**. [s.l: s.n.].

KHALID, O. et al. **OmniSuggest: A Ubiquitous Cloud-Based Context-Aware Recommendation System for Mobile Social NetworksIEEE Transactions on Services Computing**, 2014.

KIRKHAM, R. et al. **The break-time barometer - An Exploratory System for Workplace Break-time Social Awareness**. Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing - UbiComp ’13. **Anais**...: UbiComp ’13.New York, New York, USA: ACM Press, 2013Disponível em: <http://doi.acm.org/10.1145/2493432.2493468>. Acesso em: 13 ago. 2014

KOEHLER, C. et al. **TherML: Occupancy Prediction for Thermostat Control**. Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing. **Anais**...: UbiComp ’13.New York, NY, USA: ACM, 2013

KOREN, Y.; BELL, R.; VOLINSKY, C. Matrix Factorization Techniques for Recommender Systems. **Computer**, v. 42, n. 8, p. 30–37, ago. 2009.

LI, K.; DU, T. C. Building a targeted mobile advertising system for location-based services. **Decision Support Systems**, v. 54, n. 1, p. 1–8, dez. 2012.

LIU, C.; WHITE, R. W.; DUMAIS, S. **Understanding Web Browsing Behaviors Through Weibull Analysis of Dwell Time**. Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval. **Anais**...: SIGIR ’10.New York, NY, USA: ACM, 2010

LOQUES, O.; SZTAJNBERG, A. **Adaptation Issues in Software Architectures of Remote Health Care Systems**. Proceedings of the 2010 ICSE Workshop on Software Engineering in

Health Care. **Anais**...: SEHC ’10.New York, NY, USA: ACM, 2010

LUCKE, U.; RENSING, C. A survey on pervasive education. **Pervasive and Mobile Computing**, v. 14, n. 0, p. 3–16, out. 2014.

MACHADO, A. et al. **Situation-awareness as a Key for Proactive Actions in Ambient Assisted Living**. Proceedings of the 15th International Conference on Enterprise Information Systems. **Anais**...SciTePress - Science and and Technology Publications, 2013

MACHADO, G. M.; DE OLIVEIRA, J. P. M. **Context-aware adaptive recommendation of resources for mobile users in a university campus**. 2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). **Anais**...IEEE, out. 2014

MAEKAWA, T. et al. Context-aware Web Search in Ubiquitous Sensor Environments. **ACM Trans. Internet Technol.**, v. 11, n. 3, p. 12:1--12:23, fev. 2012.

MAIA, M. E. F. et al. **Framework for Building Intelligent Mobile Social Applications**. Proceedings of the 27th Annual ACM Symposium on Applied Computing. **Anais**...: SAC ’12.New York, NY, USA: ACM, 2012

MANOUSELIS, N. et al. Recommender Systems in Technology Enhanced Learning. In: RICCI, F. et al. (Eds.). . **Recommender Systems Handbook SE - 12**. [s.l.] Springer US, 2011. p. 387–415.

MARCHIONINI, G. Exploratory Search: From Finding to Understanding. **Commun. ACM**, v. 49, n. 4, p. 41–46, abr. 2006.

MARILZA PERNAS FLEISCHMANN, A. **Sensibilidade à Situação em Sistemas Educacionais na Web**. [s.l.] Universidade Federal do Rio Grande do Sul, 2012.

MAYBURY, M. T.; BRUSILOVSKY, P. From adaptive hypermedia to the adaptive web. **Communications of the ACM - The Adaptive Web**, v. 45, n. 5, p. 30–33, 2002.

METTOURIS, C.; PAPADOPOULOS, G. A. Ubiquitous recommender systems. **Computing**, v. 96, n. 3, p. 223–257, 8 out. 2013.

MOEBERT, T.; ZENDER, R.; LUCKE, U. A Generalized Approach for Context-Aware

Adaptation in Mobile E-Learning Settings. In: PEÑA-AYALA, A. (Ed.). . Cham: Springer International Publishing, 2016. p. 23–53.

NEVES, A. R. DE M.; CARVALHO, Á. M. G.; RALHA, C. G. Agent-based architecture for context-aware and personalized event recommendation. **Expert Systems with Applications**, v. 41, n. 2, p. 563–573, 1 fev. 2014.

NOGUERA, J. M. et al. A mobile 3D-GIS hybrid recommender system for tourism. **Information Sciences**, v. 215, n. 0, p. 37–52, 15 dez. 2012.

OTEBOLAKU, A. M.; ANDRADE, M. T. Context-aware media recommendations for smart devices. **Journal of Ambient Intelligence and Humanized Computing**, v. 6, n. 1, p. 13–36, 16 fev. 2015.

PARATE, A. et al. **Practical Prediction and Prefetch for Faster Access to Applications on Mobile Phones**. Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing. **Anais**...: UbiComp ’13.New York, NY, USA: ACM, 2013

PERNAS, A. M. et al. Enriching adaptation in e-learning systems through a situation-aware ontology network. **Interactive Technology and Smart Education**, v. 9, n. 2, p. 60–73, 6 ago. 2012.

PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: An update. **Information and Software Technology**, v. 64, p. 1–18, ago. 2015.

PUTHIYA PARAMBATH, S. A.; USUNIER, N.; GRANDVALET, Y. **A Coverage-Based Approach to Recommendation Diversity On Similarity Graph**. Proceedings of the 10th ACM Conference on Recommender Systems. **Anais**...: RecSys ’16.New York, NY, USA: ACM, 2016

RecSys ’16: Proceedings of the 10th ACM Conference on Recommender Systems. New York, NY, USA: ACM, 2016

RIBEIRO, F. R.; SANTOS, P.; METRÔLHO, J. Using Individual Interactions to Infer Group Interests and to Recommend Content for Groups in Public Spaces. In: ROCHA, Á. et al. (Eds.). . Cham: Springer International Publishing, 2014. p. 257–267.

RIENER, A. et al. Traffic Flow Harmonization in Expressway Merging. **Personal Ubiquitous Comput.**, v. 17, n. 3, p. 519–532, 2013.

RUOTSALO, T. et al. SMARTMUSEUM: A mobile recommender system for the Web of Data. **Web Semantics: Science, Services and Agents on the World Wide Web**, v. 20, n. 0, p. 50–67, maio 2013.

SALMAN, Y. et al. **A Proactive Multi-type Context-Aware Recommender System in the Environment of Internet of Things**. 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing. **Anais**...IEEE, out. 2015

SANTOS, P.; RIBEIRO, F. R.; METROLHO, J. **Using pervasive computing technologies to deliver personal and public ads in public spacesInformation Systems and Technologies (CISTI), 2012 7th Iberian Conference on**, 2012.

SCHAUB, F. et al. **Privacy Context Model for Dynamic Privacy Adaptation in Ubiquitous Computing**. Proceedings of the 2012 ACM Conference on Ubiquitous Computing. **Anais**...: UbiComp ’12.New York, NY, USA: ACM, 2012

SCHEDL, M. **Ameliorating Music Recommendation: Integrating Music Content, Music Context, and User Context for Improved Music Retrieval and Recommendation**. Proceedings of International Conference on Advances in Mobile Computing & Multimedia. **Anais**...: MoMM ’13.New York, NY, USA: ACM, 2013

SCHEDL, M.; BREITSCHOPF, G.; IONESCU, B. **Mobile Music Genius: Reggae at the Beach, Metal on a Friday Night?** Proceedings of International Conference on Multimedia Retrieval. **Anais**...: ICMR ’14.New York, NY, USA: ACM, 2014

SHABIB, N.; KROGSTIE, J. **The Use of Data Mining Techniques in Location-based Recommender System**. Proceedings of the International Conference on Web Intelligence, Mining and Semantics. **Anais**...: WIMS ’11.New York, NY, USA: ACM, 2011

SHAO, J.; GREENHALGH, C. **DC2S: A Dynamic Car Sharing System**. Proceedings of the 2Nd ACM SIGSPATIAL International Workshop on Location Based Social Networks. **Anais**...: LBSN ’10.New York, NY, USA: ACM, 201

SILVA, L. et al. Recommendation of Learning Objects in an Ubiquitous Learning Environment through an Agent-Based Approach. In: PUTNIK, G.; CRUZ-CUNHA, M. (Eds.). . **Virtual and Networked Organizations, Emergent Technologies and Tools SE - 11**. Communications in Computer and Information Science. [s.l.] Springer Berlin Heidelberg, 2012. v. 248p. 101–110.

SMIRNOV, A. et al. Smart Space-Based Tourist Recommendation System. In: BALANDIN, S.; ANDREEV, S.; KOUCHERYAVY, Y. (Eds.). . Cham: Springer International Publishing, 2014. p. 40–51.

SPEDALIERI, A. et al. **My Personal Media Entertainer: Context-adaptive Content Recommendation and Delivery**. Proceedings of the 2010 ACM Workshop on Social, Adaptive and Personalized Multimedia Interaction and Access. **Anais**...: SAPMIA '10.New York, NY, USA: ACM, 2010

TEO, C. H. et al. **Adaptive, Personalized Diversity for Visual Discovery**. Proceedings of the 10th ACM Conference on Recommender Systems. **Anais**...: RecSys '16.New York, NY, USA: ACM, 2016

TONELLA, P. et al. Empirical studies in reverse engineering: state of the art and future trends. **Empirical Software Engineering**, v. 12, n. 5, p. 551–571, 2007.

VERBERT, K. et al. **Context-Aware Recommender Systems for Learning: A Survey and Future ChallengesLearning Technologies, IEEE Transactions on**, 2012.

VICARI, R. M. et al. Proposta brasileira de metadados para objetos de aprendizagem baseados em agentes (obaa). **RENOTE: revista novas tecnologias na educacao**, v. 8, n. 2, p. 1–10, 2010.

WANG, S.-L.; WU, C.-Y. Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system. **Expert Systems with Applications**, v. 38, n. 9, p. 10831–10838, set. 2011.

WANG, X.; ROSENBLUM, D.; WANG, Y. **Context-aware Mobile Music Recommendation for Daily Activities**. Proceedings of the 20th ACM International Conference on Multimedia. **Anais**...: MM '12.New York, NY, USA: ACM, 2012

WOERNDL, W. et al. Logging User Activities and Sensor Data on Mobile Devices. In:

ATZMUELLER, M. et al. (Eds.). . **Analysis of Social Media and Ubiquitous Data SE - 1**. Lecture Notes in Computer Science. [s.l.] Springer Berlin Heidelberg, 2011. v. 6904p. 1–19.

WONG, R. K.; CHU, V. W.; HAO, T. Online Role Mining for Context-aware Mobile Service Recommendation. **Personal Ubiquitous Comput.**, v. 18, n. 5, p. 1029–1046, 2014.

YAO, L. et al. Things of Interest Recommendation by Leveraging Heterogeneous Relations in the Internet of Things. **ACM Trans. Internet Technol.**, v. 16, n. 2, p. 9:1--9:25, mar. 2016.

APPENDIX <SATISFACTION SURVEY>

Adaptive Recommender System Survey

This survey intents to collect your opinion about the system and the recommendations, it should take about 3 minutes to answer.

* Required

1- Which research group are you from? *

- ○ PPGC-UFRGS
- ○ PPGIE-UFRGS
- ○ INF-UFRGS
- ○ UDESC
- ○ Other: ____

2- What did you think about the learner profile? *

- ☐ Very good
- ☐ Good enough
- ☐ A little long to fill in
- ☐ A little intrusive
- ☐ A little confuse
- ☐ Other: ____

3- In a scale from 1 to 5, how satisfied were you with the way the movies were organized and presented in the system? *

	1	2	3	4	5	
Completely Unsatisfied	○	○	○	○	○	Completely Satisfied

4- How satisfied were you with your recommendations? *

	1	2	3	4	5	
Completely Unsatisfied	○	○	○	○	○	Completely Satisfied

5- How satisfied were you with the recommendation of similar movies? *

	1	2	3	4	5	
Completely Unsatisfied	○	○	○	○	○	Completely Satisfied

6- Did you find interesting to have a list of already seen movies? *

- ○ Yes
- ○ No

7- Please, give a rate to the whole system experience: *

	1	2	3	4	5	6	7	8	9	10	
Very bad experience	○	○	○	○	○	○	○	○	○	○	Excellent experience

8- If you want, leave some suggestions and improvements we could implement in the system (you can write both in english or portuguese):

Your answer

Thank you for your collaboration =)

SUBMIT