Cyprus University of Technology

Faculty of Engineering and Technology

Doctoral Dissertation

Designing and Evaluating Intelligent Context-Aware Recommender Systems: Methodologies and Applications

Panayiotis Christodoulou

Limassol, December 2017

CYPRUS UNIVERSITY OF TECHNOLOGY

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING, COMPUTER ENGINEERING AND INFORMATICS

Doctoral Dissertation

Designing and Evaluating Intelligent Context-Aware Recommender Systems: Methodologies and Applications

Panayiotis Christodoulou

Limassol, December 2017

Approval Form

Doctoral Dissertation

Designing and Evaluating Intelligent Context-Aware Recommender Systems: Methodologies and Applications

Presented by

Panayiotis Christodoulou

Supervisor: Andreas Andreou, Professor, Cyprus University of Technology

Signature ______________________

Member of the committee: Vassos Soteriou, Associate Professor, Cyprus University of Technology

Signature Β. Σωτηρίου ______________________

Member of the committee: Vassilakis Costas, Professor, University of Peloponnese

Signature ______________________

Cyprus University of Technology

Limassol, December 2017

3

Copyrights

Achnowledgements: Obtaining a PhD degree is an interesting journey full of new discoveries and surprises along the way. Although for me this was not easy, the knowledge acquired along this way is completely worthy. Throughout this journey I never felt alone, thus I would like to express my sincere gratitude to those who helped me to achieve my goal. I had the honor of being supervised by Professor Andreas Andreou, his constant support, guidance, friendship, patience and encouragement always kept me on the right path. Also, I am really grateful to the Associate Professor Soterios Chatzis for his collaboration, support and help. Moreover, I would like to thank my colleagues Constantinos Stylianou, Andreas Christoforou and Charis Partaourides for their collaboration. My boundless love and appreciation goes to my wife Christiana Lambrianidou and my son, for their constant love and support through this journey. Finally, I would like to thank my parents Andreas Christodoulou and Theodosia, my brother Klitos Christodoulou and all my friends for supporting me during the more stressful periods and for encouraging me to pursue my interests.

ABSTRACT

This research introduces new concepts and methodologies for Recommender Systems aiming to enhance the user experience and at the same time to improve the system's accuracy by dealing with the challenges of RS. The thesis and the corresponding research is structured in three main parts. The first part of this thesis concentrates more on the development of new Multi-criteria RS to improve the accuracy and performance of RS. Our study examines solutions on how to deal with data sparsity, scalability issues and the cold-start problem by utilizing various techniques. The second part deals with the classification prediction problem. We propose a new methodology for developing hybrid models to improve the accuracy of classification models and thus provide better recommendations. The final part introduces a Recurrent Latent Variable framework based on a variational Recurrent Neural Network that deals with data sparsity and uncertainty met on session-based recommendations and sequence-based data. Experimentation was performed in all three parts mentioned and the results demonstrated the validity of the proposed methodologies when compared with state-of-the-art methods.

Keywords: Multi-criteria Recommender Systems, Recommendations utilizing classification models, Session-based recommendations, Sequence-based data

PUBLICATIONS

In the context of this thesis the following papers have been produced and published:

1. A dynamic Web Recommender System using Hard and Fuzzy K-modes clustering

   Christodoulou Panayiotis, Lestas Marios & Andreou S. Andreas

   9th IFIP WG 12.5 International Conference on Artificial Intelligence Applications and Innovations, 2013, Paphos, Cyprus

2. Applying hard and fuzzy K-modes clustering for dynamic Web recommendations

   Christodoulou Panayiotis, Lestas Marios & Andreou S. Andreas

   Engineering Intelligent Systems, 2014, Volume 22, Issue 3-4, Pages 177-190

3. A Real-Time Targeted Recommender System for Supermarkets
   Christodoulou Panayiotis, Christodoulou Klitos & S. Andreou Andreas
   19th International Conference on Enterprise Information Systems (ICEIS)

4. A Hybrid Prediction Model Integrating Fuzzy Cognitive Maps With Support Vector Machines
   Christodoulou Panayiotis, Christoforou Andreas & Andreou S. Andreas
   19th International Conference on Enterprise Information Systems (ICEIS)

5. Improving the Performance of Classification Models with Fuzzy Cognitive Maps
   Christodoulou Panayiotis, Christoforou Andreas & Andreou S. Andreas S.
   2017 IEEE Conference on Fuzzy Systems

6. A Variational Recurrent Neural Network for Session-Based Recommendations using Bayesian Personalized Ranking
   Christodoulou Panayiotis, Sotirios P. Chatzis & Andreou S. Andreas
   26th International Conference on Information Systems Development

7. Recurrent Latent Variable Networks for Session-Based Recommendation
   Christodoulou Panayiotis, Sotirios P. Chatzis & Andreou S. Andreas
   2nd Deep Learning technology for Recommender Systems - RecSys 2017

8. A Variational Latent Variable Model with Recurrent Temporal Dependencies for Session-Based Recommendation (VLaReT)
   Christodoulou Panayiotis, Sotirios P. Chatzis & Andreou S. Andreas
   Information Systems Development Methods, Tools and Management, Lecture Notes in Information Systems and Organisation

9. A Recurrent Latent Variable Model for Supervised Modeling of High-Dimensional Sequential Data (under review)
   Christodoulou Panayiotis, Sotirios P. Chatzis & Andreou S. Andreas
   Expert Systems With Applications

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CUT:	Cyprus University of Technology
RS:	Recommender Systems
CB:	Content-based
CF:	Collaborative filtering
MF:	Matrix Factorization
SVM:	Support Vector Machines
LDA:	Linear Discrimination
k-NN:	k Nearest Neighbors

Chapter 1: Introduction

Recommender Systems (RS) are intelligent engines that collect information related to what a user has previously seen or bought with the aim of providing back to the user personalized suggestions on unobserved items that are likely to be of interest. RS are classified into three broad categories: (a) *Content-based* (CB) systems, which use keywords to suggest items to a user similar to the ones preferred in the past (Konstan & Riedl, 2012); (b) *Collaborative Filtering* (CF) techniques, that recommend items to a user based on the items other users had previously seen or bought, or based on similarities that exist between the items that a user bought or seen and other items. CF systems use various strategies based on user-based, item-based, matrix factorization and clustering techniques, (Ning et al., 2015) to find correlations between users/items and produce recommendations; (c) *Hybrid Recommendation* (HR) approaches, which use a combination of CB and CF methods to deal with the limitations that exist in the aforementioned systems (Adomavicius & Tuzhilin, 2005).

CB systems provide better accuracy of results when dealing with items containing textual information; however, they face a lot of challenges. First of all, such systems lack the ability to distinguish how well a text description is written from a badly one, especially in the case when they use similar or related terms (Adomavicius & Tuzhilin, 2005). Additionally, CB systems are often limited by the *over-specialization problem*; when a system recommends items with high similarity compared to a user's profile, the user is likely to be suggested with items similar to the ones that has already seen (Lü et al., 2012). Finally, a CB system requires a significant number of ratings before suggesting items with high accuracy to a user (Lü et al., 2012) threfore, when a new user registers into the system and has few or no ratings at all, it is more likely to get low accuracy recommendations (known as the *new-user problem*) (Adomavicius & Tuzhilin, 2005).

In contrast to CB systems, CF approaches have more capabilities than the CB methods, but they can also lead to poor performances due to a number of problems. CF systems face the *data Sparsity problem* (Adomavicius & Tuzhilin, 2005): the number of items that exist on e-commerce websites is enormous; as a result, the most active users usually rate only a subset of the entire dataset. This means that many of the most popular items have few ratings and thus the likelihood of the system suggesting them is relatively low (Ning et al., 2015). Similarly, to CB systems, CF systems require a significant amount of existing data on a user profile, before being able to make accurate suggestions. In addition, new items need to be rated by a substantial number of users, otherwise the RS would not be able to come up with proper recommendations (Ning et al., 2015). Finally, RS are challenged by *scalability issues*, taking into account the vast amount of data that exists on websites and applications, a considerable amount of computational power is needed to compute accurate recommendations on time, and this is something that needs to be dealt with (Pu, Chen & Hu, 2012).

Several methodologies in the literature suggest various solutions to overcome the aforementioned limitations of RS (Cacheda et al., 2011) while focusing also at the same time on possible extensions or capabilities. *Context-aware systems* make use of various interesting contexts, such as, time, location and occasions to facilitate the recommendation process and deal with challenges (Adomavicius et al., 2011); *Multi-criteria systems* generate recommendations based on multiple-criteria techniques by modeling the usefulness of an item for a user as a vector of ratings referred to several criteria (Hwang, 2010); *Temporal dynamic systems* take into consideration user preferences, such as, tastes or interests that tend to continuously change over time (Koren, 2010); *Session-based systems* provide recommendations taking into consideration the actions of users in a current browsing session (Tan et al. 2016). The above-mentioned capabilities of RS consider only a subset of RS enhancements that can improve the accuracy, quality and performance of a RS.

Despite the advances of RS in a wider spectrum of applications, there is ample room for improvements. This thesis describes various frameworks for applying RS in real-world scenarios where users dynamically and/or continuously interact with such systems.

The following research questions motivated this research. RQ1 – How does a RS deal with the fundamental problem of data Sparsity, the cold-start problem and scalability issues that exist in a real-world application? RQ2 – How accurate and personalized are the recommendations provided by a multi-criteria system? RQ3 – How well machine learning techniques can improve the recommendation process? RQ4 – How deep learning models can deal with data sparsity and uncertainty that exist in session-based recommendations/sequence-based data?

For seeking answers to the aforementioned questions, this thesis contributes the following: (a) Examines the use of an entropy-based approach and Hard and Fuzzy K-modes clustering algorithms to overcome the cold-start and data sparsity limitations; (b) Studies the adoption of a rule-based system to discover sub-datasets of items in order to face scalability issues and assist the recommendation process; (c) Inspects the use of a Bayesian Inference model, that is trained on users dynamic information to predict whether an item is likely to be recommended or not; (e) Examines the employment of recommendation engines in real-time environments to suggest the *top-k* personalized items to users; (f) Studies the use of machine learning techniques in combination with Fuzzy Cognitive Maps to improve the system's accuracy and finally, (g) Implements deep learning models to deal with data sparsity and uncertainty on session-based/sequence-based recommendations.

The first part of our research concentrates more on the development of new multi-criteria systems aiming to improve the accuracy and performance of RS by utilizing various techniques. Our research examines solutions on how to deal with data sparsity, scalability issues and the cold-start problem. A new tool was developed that maps different datasets and converts a dataset into numeric values in order to overcome scalability issues and reduce the computational power needed when producing the recommendations. Moreover, the use of an entropy-based algorithm that utilizes a similarity measure to find the appropriate number of clusters and the corresponding centers based on any dataset, as well as the use of Hard or Fuzzy K-modes clustering algorithms to group together users/items with similar characteristics help the proposed system to deal with the cold-start problem and the data sparsity challenge. Users/items belong in specific clusters so when the recommendation engine produces recommendations it looks only at the specific cluster where a user/item belongs to thus reducing processing time. New users are also fit into a cluster so the proposed methodology can suggest items back to them even if they did not rate or see a certain number of items. Furthermore, even the most popular items that have few ratings belong to a cluster; thus they can now be recommended by the system dealing with the data sparsity challenge. Finally, the adoption of rule-based systems aims at facing the scalability concerns of RS by applying specific rules which help the system minimizing the dataset needed to compute recommendations. Multi-criteria systems are mainly focused on user profiles that carry information about the user's interests, preferences and tastes that are exported either explicitly or implicitly.

To this end, we firstly developed a movie Recommender system that makes recommendations based on the preferences of the interested user, which are dynamically changing by taking into consideration his/her searches in real-time. This approach is enhanced by the utilization of static preferences which are declared by the user when registering into the system. The clustering procedure, which is the heart of the recommendation engine, is of particular importance and a number of the techniques mentioned above, such as Entropy-Based, Hard K-modes and Fuzzy K-modes, are utilized in order to cluster users/items. The proposed system was tested using the MovieLens1M dataset that was linked with IMDB.com to retrieve more content information regarding items. The results indicated that the proposed methodology meets the design objectives as it delivers items which are closely related to what he/she would have liked to receive based on how the user ranked the different categories and based on his/her previous behavior. An extension of this system was developed in order to strengthen our methodology. Unlike the algorithm followed in the first approach, the new method is not enhanced by the utilization of static preferences declared by the user when registering into the system, but it now relies on a learning mode for new registered users. According to this process, the system records the users' preferences and a number of searches (learning period) and then it starts recommending items. The updated proposed system was tested on the same dataset as before also linked with IMDB.com but now it retrieved content information related not only to the movie categories but also to the movies' stars and production companies. For experimentation purposes users were searching items based on stars, categories, production companies and any combination between them. The final results suggested that the proposed system produces recommendations that are closely related to the user's preferences.

Under the same context we also introduced a new framework for deploying a Recommender System in a store environment aiming at suggesting real-time personalized offers to customers. Store customers find it difficult to choose from a large variety of products or be informed for the latest offers that exist in a shop based on the items that they wish to purchase. Under this context we implemented a RS where as customers navigate in a store various mechanisms such as iBeacons push personalized notifications to their smart-devices informing them about the latest offers that are likely to be of interest. Notifications are also sent to a customer's smartphone using mail servers or SMS. The proposed methodology is again using an Entropy-based approach to determine the number of clusters and the clusters' centers from a dataset of registered users with different preferences (static information) and a Hard K-modes clustering algorithm to group users with similar characteristics in different clusters based on their preferences. As already described before, these methodologies help us to deal with the cold-start problem and tackle scalability issues. Furthermore, after grouping users into clusters, a rule-based system is applied to create personalized sub-datasets of products for each cluster. This helps reducing the search space in the overall set of products and deal with data sparsity as the system now produces recommendations by utilizing only specific sub-datasets of products and not the overall dataset. For producing suggestions, we use a probabilistic model that utilizes the users' transaction histories in order to learn their frequent shopping habits and then use this information as input to a Bayesian Inference approach to determine whether a product that is on offer is suitable for purchase or not. The proposed recommendation engine described in this part suggests targeted products from a list of offers to users in real-time while they are navigating into a store without spending time on the shelves, and it was tested on real-world and synthetic data. The final results indicated an increase on the system's accuracy when compared with classic collaborative filtering methods.

The second part of this study deals with the classification prediction problem. Prediction is a vital issue that applies in every scientific discipline and is considered as a problem that involves multiple factors. In some cases, the prediction process can be described as highly complex exhibiting high levels of uncertainty. Researchers have now been focused on the two major aspects of prediction models and struggle to develop tools and approaches that can actually provide accurate results. Accuracy and time are the two important aspects of prediction as they characterize a model's performance. Aiming to tackle the aforementioned challenges this part introduces a series of new hybrid prediction models that exploit the advantages offered by Fuzzy Cognitive Maps (FCMs) coupled with the prediction abilities of classic classification models such as Support Vector Machines, Linear Discrimination, Classification Trees and the weighted *k*-nn approach in order to produce methodologies that are able to provide accurate predictions. Final results indicated that the proposed hybrid models deliver accurate results outperforming traditional classification models.

The final part of our research deals with session-based recommendations and sequence-based data. Session-based recommendation is a recent challenge in the area of Recommender Systems and it was first introduced in the RecSys Challenge of 2015 (Hidasi et al., 2015). In this context a system provides recommendations taking into consideration only the actions of users in the current browsing session. This kind of recommendations processes the historical data of users that are captured during an active session and relies only on a narrow piece of information that describes the behavior of a specific user. The goal of such systems is to predict the user's next move in a session in order to produce accurate recommendations. In the session-based recommendation problem the RS considers the first item that a user clicks/views, when accessing a website, as the initial input of a Recurrent Neural Network (RNN) and then every other sequential click produces a recommendation (output) that relies upon the previous clicks. The main challenges in these systems are the set of items to choose from, the training time of the model and the scalability issues due to the fact that the click-stream datasets are enormous. In order to deal with the aforementioned challenges our research introduces a variation of the classic RNN approach. This part presents a Recurrent Latent Variable framework for Session-Based Recommendations that utilizes a Bayesian Personalized Ranking aiming to increase the benefits of a session-based RS when dealing with data sparsity and uncertainty. The proposed methodology is inspired by the recently proposed systems that utilize Bayesian inference techniques. In a RS environment a Bayesian inference considers the system variables with some prior distribution on them; this helps the recommendation engine to deal with uncertainty over the sparse data and produce improved predictive results. The performance of the proposed model was compared against state-of-the-art techniques and the results showed that our methodology performs better without suffering from scalability issues. Furthermore, using the same methodology we attempt to ameliorate the impact of data sparsity in the context of supervised modeling applications dealing with high-dimensional sequential data. Specifically, the proposed model is capable of extracting subtle and complex underlying temporal dynamics in the observed sequential data, so as to inform the predictive algorithm. We evaluate the efficacy of the so-obtained approach, considering challenging publicly available benchmarks, dealing with diverse application areas. As we empirically demonstrate, our approach completely outperforms the competition, without presenting any limitations in terms of computational efficiency and scalability.

It should be noted that in the experimental part of each of the methods introduced in this thesis the comparisons with other methods/models in literature was performed by reexecuting them reproducing the corresponding results.

The remainder of this thesis is structured as follows: Section 2 presents an overview on Recommender Systems by presenting the various categories and challenges of RS while Section 3 describes their capabilities, trends and recent challenges when applied in various contexts. Sections 4,5 and 6 introduce new concepts and methodologies for RS aiming to enhance user experience and at the same time improve the system’s accuracy by dealing with challenges faced by RS. More specifically Section 4 outlines our work on multi-criteria systems applied in various contexts, Section 5 introduces hybrid models for prediction and recommendation purposes and Section 6 presents a novel model to deal with session-based recommendations on sequence-based data. Finally, Section 7 concludes this thesis with a brief overview of its basic findings as well as future research steps.

Chapter 2: Overview of Recommender Systems

2.1 Introduction

The success of the World Wide Web in early nineties led to the development of e-Commerce websites consisting with thousands of products; as the number of the products on those websites was increasing the need for tools able to make the user's life easier was deemed necessary. RS firstly appeared as tools that could provide solutions to the aforementioned problem and since the introduction of the first scientific papers in the nineties RS has become a crucial research area (Ma et al. 2011).

RS can be seen as smart search engines that collect information about users or items aiming to provide customized suggestions back to the users by utilizing various techniques. RS are capable of recommending items back to a user based on his/her preferences on previously bought items, or based on a user's preferences on specific items that he/she is looking at compared to similar users, or, based on keywords similarity between a certain item and others. RS utilize user profiles that carry information about the user's interests, preferences and tastes and are exported from users either explicitly or implicitly. Explicit ratings are recorded when a user is providing an opinion about an item either by scalar ratings or surveys. Implicit ratings are the user's actions that are recorded while navigates on a website or application in real-time (Schafer et al., 2007). Ratings in RS can take a variety of forms: scalar ratings (1-10) or ordinal ratings (agree, disagree), binary ratings (good/bad) and finally unary ratings (observations or bought items) (Schafer et al., 2007).

RS are classified into three broad categories. Content-based systems (CB), Collaborative filtering techniques (CF) and Hybrid models. CB methods use keywords and suggest items to a user comparable to those the user liked in the past. CF approaches are grouped into two major categories: The Memory-based methods and the Model-based methods (Adomavicius & Tuzhilin, 2005). The Memory-based methods are divided into two types: the non-personalized methods and the neighbourhood models (personalized). The non-personalized approaches produce predictions based on the entire set of ratings without the need of a users profiles while the personalized techniques recommend items to a user based on the items the user has seen or bought in the past, or based on the similarities between the active user and other users' who share similar tastes/interests. CF methods can also utilize various approaches in order to train a model (learning process) and then use that model to make rating predictions on real data; these techniques are call Model-based. Hybrid approaches consider to be the last main category of RS, more specifically Hybrid methods are a combination of separate CB and CF systems.

This section first discusses the basic concepts of the RS techniques and their main challenges and then presents any related work conducted on this topic to overcome those challenges. In addition, this section describes various extension capabilities of RS that can lead to more accurate recommendation results and, finally, it presents several metrics that are used to measure the similarity between users/items or the accuracy and performance of a system.

2.2 Basic Concepts of RS

2.2.1 Content-Based RS

2.2.1.1 Overview of CB systems

The Content-Based approach has its origins in information retrieval and filtering. Most of the CB systems are focused on suggesting items containing keywords. Numerous algorithms for automatically assigning categories to articles and measuring the similarity between documents aiming to suggest the most related articles to the active user were developed in the last decades (Adomavicius & Tuzhilin, 2005). The Naïve Bayesian classifier is the most famous probabilistic text classification approach. It aims to assign categories to text documents (Lü et al., 2012). The term frequency/inverse document frequency measure (TF-IDF) is also one of the best-known measures for setting keyword weights in Information Retrieval (Adomavicius & Tuzhilin, 2005). An additional method used to weight keywords in CB systems is the Rocchio algorithm that utilizes analysis techniques on keywords to compute a user's profile as an average vector from individual content vectors. Users' profiles in these systems are retrieved by analyzing the substance of the items that a user have previously seen or rated (Pazzani & Billsus, 2007). Cosine similarity is a different metric which measures the cosine of the angle between items and uses vectors for expressing the similarity between a user's profile and the content. For example, assume that a user reads many online articles for computers. The CB system will be able to suggest other computing articles back to the user only if these articles have more computing-related terms than other articles on separate topics (Lü et al., 2012). A RS with cosine similarity measure will assign a greater rating to those articles that have high-weighted computer terms and lower scoring to the ones where computing terms are less weighted.

2.2.1.2 Challenges of CB systems

CB systems face several challenges due to limited content analysis (Adomavicius & Tuzhilin, 2005). Informational retrieval procedures work well only in extracting features from text documents. Other domains such as videos, images, sounds have an inherent problem with automatic feature extraction. Another CB system limitation is that these systems cannot recognize a well-written article from a badly written one if the articles use the same terms since the documents are regularly represented by their most critical keywords. Moreover, the CB systems suffer from the over-specialization problem. If the system recommends items that have a high score compared to a user profile, the user faces the difficulty of being continuously suggested items similar to those he/she already rated or seen (Lops, De Gemmis & Semeraro, 2011). CB systems have the limitation of not recommending items that have many variations from the documents that a user has already seen. A last challenge of the CB approaches is the new user problem (cold-start problem) (Lü et al., 2012). A newly registered user that has few or null ratings would not be able to get accurate recommendations. CB systems require from the the user to rate a sufficient number of items before it can understand the user’s preferences and suggest back accurate results.

2.2.1.3 Dealing with challenges of CB systems

This section provides different studies from the literature that managed to overcome the limitations of CB systems. The authors in (Cantador et al., 2010) evaluate multiple CB recommendation approaches that use user and item profiles described in complete lists of social tags. Results show that the proposed models focused on user profiles outperform the models oriented to item profiles. The BM25 algorithm introduced in this study performed better than the standard TF-IDF because in a tagging system the most popular tags are punished more correctly.

The CB recommendation method presented in (Chei et al. 2010) is trying to achieve a better user attention using data from Twitter. Twelve algorithms were executed to identify the content resources, topic interests and social voting on those data samples. The best performing algorithm, namely, FoF-Self-None improves the ratio of better content significantly when compared with other approaches. Another system also tested on Twitter was the one proposed by the authors in (Hannon et al., 2010); this CB system focuses more on the creation of relationships between users on a social network and demonstrates the potential for effective and practical followee recommendations.

The authors in (Li et al., 2010) present a CB system and argue that by combining the features of news recommendations and the elements that exist from user interactions the over-specialization problem can be addressed. Finally, the work in (Mooney et al., 2000) describes a CB recommender system that utilizes information extraction and a machine-learning algorithm in order to deal with text categorization; the system suggests unrated items to users with unique interest providing at the same time accurate recommendations and explanations for the recommendations.

2.2.2 Collaborative Filtering

2.2.2.1 Memory-Based CF approaches

Memory-based methods do not take into account the fact that various users may use the rating scale in a different way and use heuristics algorithms that make predictions based on the entire collection of items that was previously rated by the user. These systems include also techniques that do not make use of a user profile. Non-personalized methods present to a user a predefined list of recommendations despite the user's preferences. The top-N is a non-personalized method that recommends items back to the active user that have the largest average rating on the overall dataset. Some other non-personalized approaches are the top-Popular method that recommends to the user the top-N items with the largest number of ratings and the top-k hit method that suggests to the user the items with the highest number of clicks. Except from the non-personalised methods, memory-based CF systems utilize models that take into consideration a user's profile and base their prediction on similarity correlations between users or items. User-based similarity methods predict the rating for an item for a user based on the ratings exposed by similar users, namely neighbors. Item-based similarity methods compute the user preference for an object based on the user's ratings on similar items (Lü et al., 2012).

Numerous methods for finding and computing the similarities between users and items have been examined in the literature. The Pearson correlation coefficient is one of the most successful approaches for finding the similarities in a user-based CF system (Adomavicius & Tuzhilin, 2005). This metric estimates users' similarity by finding the difference between the rating of a combination of items and the average score for those items rated by similar users. Another approach that aims to find the similarity between users is the Cosine-based metric also knows as Vector similarity (Lü et al., 2012). In the Cosine-based process, the similarity between two users (vectors) is calculated by finding the angle's cosine. Moreover, there are several other approaches for computing similarity. The Mean Squared Difference (MSD) (Lü et al., 2012) computes the similarity between users based on the mean difference of the items that both users have rated; the users who have higher difference than a certain threshold are rejected and the similarity of the rest is weighted. The Weighted Pearson measure finds the confidence placed on a neighbour and increases as the number of items in common between users' increases (Adomavicius & Tuzhilin, 2005). The Correlation threshold is a method for choosing a number of neighbors for a user when the neighbors' similarity exceeds a given value. A last approach for selecting neighbors is the max number of neighbors' method that selects, after measuring the similarities, a number of users that are most similar to the actual user (Park et al., 2012).

In the following we present some metrics used in item-based methods. In these methods the prognostication for an item is based on the active user's personal ratings on similar objects (Lü et al., 2012). The main advantage of the item-based methods over the user-based methods is that the items' similarity looks more static than users' similarity so neighbourhood objects can also be determined offline. The Adjusted Cosine Similarity measure is one of the most general approaches for finding the similarities in an item-based CF method (Konstan & Riedl, 2012) and is taking into account the difference in a rating scale between the different users. It is a transformation of the Vector-based similarity and considers that users have different rating schemes. Some users may rate items highly in general while others may give to items lower ratings. In order to remove this disadvantage from the Vector-based similarity, the average ratings for a particular user are deducted from each user's rating for that specific pair of items. Another popular approach that is used in item-based CF is the Weighted Sum (Konstan & Riedl, 2012); this method takes at first all the items similar to the target item and selects the items that the user has rated; then it weights the user rating on each one of these items in order to obtain the similarity between an item and the target item.

Finally, there are approaches that make use of both users and items. Similarity Fusion is a method that combines both users and items and aims to achieve more accurate recommendations when few ratings are known (Lü et al., 2012). This approach combines together the item ratings of all users that are similar to the current user, the user's ratings on similar items and finally the ratings of similar items provided by related users to suggest accurate recommendations.

2.2.2.2 Model-Based CF approaches

Model-based CF methods utlize a collection of ratings to train a model (learning process) and then use that model to make predictions on real data. Various approaches that are using a certain model in order to recommend items are presented below. Bayesian Networks is a popular technique which represents each item as a node where the node states correspond to the potential rating values for every item. In general, this method reflects the states of a part of the world that is being modelled and describes how these states are related by probabilities. In these techniques data is used for learning the networks' structure and the conditional probabilities (Adomavicius & Tuzhilin, 2005). Clustering models like the K-means or K-modes group a set of objects in such a way that the items in the corresponding cluster are more similar to each other than to those in separate clusters (Lü et al., 2012). Except from the Bayesian and clustering methods, there are models that express users and items as vectors in the same latent factor space using hidden factors (Lü et al., 2012). In such models, the rating of a user on an item is predicted by the closeness between the rated latent factor analysis. Several RS algorithms have been proposed in literature which are based on latent factor models. Most of them factorize the user-item rating matrix and are known as Singular Value Decomposition (SVD). The idea behind SVD is to factorize an m by n matrix X into three matrices. The dataset in this model is expressed as a matrix where the rows represent the users, the columns represent the objects and the personal records are the individual ratings. Moreover, in order to have a baseline, all the blank cells are filled with the average rating for that specific item and the SVD is calculated. Other models based on SVD include the Regularized SVD, in which each item is expressed by a set of features and each user as a set of values indicating the user's preference for the various aspects of the items (Adomavicius & Tuzhilin, 2005). The values of such vectors are estimated by the model using a variation of the SVD where the unknown ratings are ignored. The Regularized SVD is much faster than the classic SVD method because it simplifies the computation process. Finally, the Matrix factorization (MF) techniques, which are also model-based CF approaches, are more efficient than the SVD because they can discover the underlying features between the users and items. MF approaches factorize the user-item rating matrix in a product of two lower rank matrices one containing the user factors and the other one the item factors, and then it models users and items as vectors in the same latent factor space (Park et al., 2012). In such a space users and items are directly comparable.

2.2.2.3 Challenges of CF methods

RS are used as effective tools that help users in finding new items they might be of interest to them (Schafer et al., 2007). Although their aim is to provide accurate recommendations back to the user, they come along with many challenges that need to be addressed. Data sparsity (Huang et al., 2004) is one of the main challenges of CF techniques faced by many e-commerce websites. The number of items sold on major websites is enormous; as a result, the most active users just rate a tiny subset of the whole dataset of products, therefore even the most popular items have few ratings (Pu et al., 2012) so they can be never recommended. Moreover, CF methods face the cold-start problem. CF systems require a large amount of existing data on a user in order to make accurate recommendations. Moreover, a new item needs to be rated by a specific number of users in order for the system to suggest it. The system needs to overcome these limitations before producing recommendations. Finally, scalability is another challenge that exists in RS; due to the fact that there are millions of websites and applications and each one of them has thousands of users and products a tremendous amount of computation power is needed to compute recommendations. Users must be able to get accurate recommendations on time thus this is also a main challenge of RS (Lü et al., 2012).

2.2.2.4 Dealing with challenges of CF systems

The authors in (Cremonesi et al., 2010) handle the new users and the new rating problem using a model-based algorithm that is called PureSVD. PureSVD is an adjustment of the SVD method; it represents users as a combination of item features offering elasticity and producing at the same time accurate recommendations to new users. The popularity and item-based algorithms presented in (Rashid et al., 2002) are used to solve the new user problem using a different approach that performs better than the classic strategies. According to the literature the use of a Matrix factorization model can also provide solutions to the cold-start problem (Gantner et al., 2010) by mapping the item attributes to the latent features of a MF model that is used to provide accurate recommendations back to the active user.

The memory-based CF approach presented in (Wang et al., 2006) deals with the problems of data sparsity and prediction quality. It treats every individual user-item ratings as predictors of missing ratings and shows that this model is doing well in terms of sparsity providing accurate recommendations back to the users. The work in (Gong, 2010) solves the structural problems of CF, namely, sparsity, scalability and cold-start, by using a K-means clustering algorithm that groups users based on their item ratings. In this approach each of the clusters has a representative center so based on the similarity between a user and the cluster center the nearest neighbors of that user are discovered which are used in the recommendation process. Moreover, another method to overcome the limitation of sparsity is discussed in (Anand & Bharadwaj, 2011); the authors propose a parameter that adjusts the weight given to global neighbors with regards to the weight given to local neighbors producing excellent recommendations. A new CF algorithm namely, Eigentaste is presented in (Goldberg et al., 2001); this method uses general queries instead of user selected queries to address the sparsity problem and outperforms other algorithms in terms of performance and accuracy. The clustering approach proposed in (Pham et al., 2011) performs better than traditional CF algorithms; this method uses the social information of users for producing recommendations in order to deal with the sparsity problem. Finally, a modified CF algorithm is introduced in (Liu, Zhou, Che, Wang & Zhang, 2010) in order to deal with sparsity problems and has a higher accuracy than the standard CF.

The work in (Karypis, 2001) addresses the scalability concerns of RS for which item-based recommendation techniques were developed. The proposed class of algorithms that utilizes cosine and conditional probability schemes uses the normalized similarities between each item to lead to more accurate recommendations than traditional results. The work in (Koren, 2010) tackles the limitations of RS by factoring the item-based and user-based approaches; the neighbourhood models are now scaling linearly with the size of the data. Finally, another method used to improve the quality of the RS is outlined in (Gao et al., 2011); this approach incorporates the weight of a user into the computation of item similarities by improving the recommendations results of the typical Adjusted Cosine and Slope one item-based method.

The classic memory-based algorithm is improved in (Jeong et al., 2010) by introducing a similarity method that utilizes an iterative message passing procedure in order to deal with the drawback of using the popular mean absolute error for performance evaluation. The authors in (Jahrer et al., 2010) and (Koren, 2008) show that by combining a set of CF algorithms (SVD, Neighboorhood, restricted Boltzmann machine) the accuracy of RS increases outperforming any single CF method. The work in (Ma et al., 2011) illustrates a novel probabilistic factor analysis model that uses the users' tastes and those of their trusted friends in massive datasets and aims at modeling RS more accurately outperforming state-of-art approaches such as UserMean, ItemMean and NMF. Moreover, by combining a Matrix factorization algorithm along with the Markov chains method as presented in (Rendle et al., 2010) outperforms the common Matrix factorization and the un-personalised Markov Chains model. The authors in (Miller et al., 2004) present a CF algorithm, namely PocketLens, that uses p2p architecture methods for finding the neighbors and produces accurate recommendations. Finally, the work in (Hofmann, 2004) describes a new family of model-based algorithms designed for CF that rely on statistical modeling techniques which introduce latent class variables to discover interest profiles. The main advantages of this technique over the standard memory-based methods are higher accuracy, constant time prediction and an explicit model representation.

According to (Bollen et al., 2010) even if people are excited by high-quality recommendation collections, the psychological study on choice overload explains why choosing an item from a set that contains many winning items can be a challenging task.

In (Bollen et al., 2010) a Matrix factorization algorithm is applied on the MovieLens dataset and used to investigate the effect of the recommendation set and the set quality on choosing items. Experimental outcomes show that bigger sets carrying only useful items do not certainly result in a superior choice pleasure compared to smaller sets.

2.2.3 Hybrid Systems

2.2.3.1 Overview of Hybrid Systems

CF methods use the hypothesis that people with similar tastes will rate things in the same way; therefore, it requires past ratings for an object to predict its suitability for a new user but they do not need content. CF systems also use the assumption that items with similiar features will be rated even. On the other hand, CB methods predict items without the need of ratings but with the need of content in order to perform its analysis process (Adomavicius & Tuzhilin, 2005). The combination of the CF and CB methods into Hybrid systems helps to overcome the challenges of data sparsity, cold-start problems and scalability issues that exist in RS. A first method for combining both systems is to implement unique CF and CB systems and then combine the ratings obtained from the individual RS into a final recommendation list using either linear combination of ratings or a voting scheme. A different approach is to add the CB characteristics into CF models. This method is based on traditional collaborative techniques but also maintains the CB profiles of each user that is used to calculate the similarity between users. The proposed approach overcomes the data sparsity problem since not many pairs of users have a significant number of likewise rated items. A final approach for combining both recommenders to a Hybrid system is to add the CF characteristics into a CB model. This method, at first, uses a dimensionality reduction technique on a group of CB profiles, and then it utilizes a latent semantic indexing to form a collection of user profiles where the users' profiles are represented by vectors. This approach results in a performance improvement in comparison to the pure CB approach (Adomavicius & Tuzhilin, 2005).

2.2.3.2 Dealing with challenges using Hybrid models

The work in (Melville et al., 2002) outlines a framework for combining CB and CF approaches to defeat the limitation that CF and CB methods in some cases fail to provide useful recommendations. This is done by using at first a CB predictor to enhance existing user data and then provide personalized suggestions through a CF system. Moreover, a CB recommendation tool used to learn the users' profiles is combined in (Liu et al., 2010) with an existing CF mechanism that generates news recommendations to help users in finding interesting articles to read. The hybrid fuzzy linguistic RS outlined in (Porcel et al., 2012) aims at helping the University's staff in the dissemination of research resources that are interesting for users outperforms other state-of-art collaborative approaches in terms of accuracy and performance. Another novel hybrid RS is the one proposed by the authos in (Su et al., 2010). The new system that is called Fusion of Rough-Set and Average-category-rating (FSRA) assists a user to recognize clearly what it prefers and not making the user confused. The proposed methodology integrates various contents and shared information to predict users' preferences. FSRA can successfully reduce the gap between user preferences and automated recommendations than any other well-known method.

A model outlined in (Schein et al., 2002) combines content and collaborative data under a single probabilistic framework to deal with the cold-start problem. A new performance metric called CROC curve is proposed that demonstrates the various components of the testing strategy that are combined to obtain a better performance on a RS. A novel music recommendation algorithm is presented in (Bu et al., 2010) that uses multiple kinds of social media information and music-based content to move away from traditional CF music recommendation websites (Cantador et al., 2010). Moreover, a hybrid system namely EntreeC is described in (Burke, 2002) that combines a knowledge-based recommendation and a CF method to recommend venues. The semantic ratings obtained from the knowledge-based part of the system enhance the effectiveness of the CF procedure producing better recommendations. The authors in (De Campos et al., 2010) describe a Bayesian network model to deal with the problem of hybrid recommendations. Bayesian networks used in the context of Artificial Intelligence (AI) have been practiced to problems with a high level of uncertainty. The effectiveness of these models helps to improve the accuracy of the proposed RS when compared with other methods.

The work of (Wang & Blei, 2011) develops an algorithm for recommending scientific articles to users for an online community by combining traditional CF methods and a probabilistic topic modeling. Results show that the proposed algorithm called Collaborative Topic Regression (CTR) outperforms the Matrix factorization and Latent

Dirichlet Allocation approaches. Additionally, the novel matrix factorization method called fLDA presented in (Agarwal & Chen, 2010) is used to predict ratings in a RS where a bag-of-words representation for item meta-data exists. In order to avoid over-fitting, the user and item factors are adjusted using a Gaussian linear regression and a Latent Dirichlet Allocation. The proposed model handles well the cold-start scenario by providing predictions with accuracy, but it also identifies at the same time interesting topics that explain the user-item interactions. Finally, the system presented in (Barragáns-Martínez et al., 2010) deals with the problem of choosing a TV program to watch; it automatically matches the user's similarities to TV programs and then suggests the ones that have the greater user preference. A SVD technique is developed to eliminate all the limitations of item-based CF. The main goal of this approach is to simplify as much as possible the user task for selecting which programs to watch on TV. Results show that low-dimension item-based filtering is very accurate and alleviates the problems of scalability and sparsity of the data.

2.2.4 Overview

This thesis examined the multiple challenges of the current recommendation methods. CB systems work well when dealing only with text documents and they cannot recognize a well-written article from a badly written one if the articles use the same terms. Additionally, if a system recommends items that have a high score compared to a user profile, the user faces the difficulty of being continuously suggested items similar to those he/she has already rated or seen and not items that have many variations from those documents, even if they are interesting to him/her (over-specialization problem). On the other hand, CF approaches suffer from the data sparsity problem. The number of items sold on major websites is enormous; as a result, the most active users will just rate a tiny subset of the whole database; therefore, even the most popular items have few ratings so they can be hardly recommended. Moreover, RS require a large amount of existing data on a user to make accurate recommendations and new objects need to be rated by a particular number of users in order for the RS to provide suggestions knows as the cold-start problem. Finally, scalability is also a challenge in the RS area due to the fact that there are millions of websites and applications, each one of them with thousands of users and products and therefore a tremendous amount of computational power is necessary to calculate recommendations. Hybrid models came as a first solution to tackle the challenges of RS but the current technological advances and the latest developments in computer science lead to methodologies that differ from the classic ones which can be applied in various contexts and solve the above-mentioned challenges in domains that we couldn’t resolved in the past. These techniques are presented in the following section.

Chapter 3: Extensions & Trends on Recommender Systems

Nowadays, the interest in the scientific area of Recommendation Systems remains high. This research area is rich of problems and currently there are only a few applications that can help users to deal with information overload and to provide accurate and reliable personalized recommendations (Ma et al., 2011), but still there is ample room for improvement. During the last decade dozens of CB, CF and Hybrid systems have been proposed while the corresponding research made an outstanding progress; nevertheless, despite all the efforts by researchers, the present generation of the systems mentioned in this section requires more extensions and enhancements (Adomavicius & Tuzhilin, 2005). This section aims firstly to identify the current trends of RS from the literature and then use the findings to decide upon the topics that this thesis will investigate.

3.1 Context-Aware Recommender Systems

Context-Aware RS (CARS) provide appropriate recommendations by adjusting them to the particular contextual situation of the user. In comparison with traditional models, CARS attempt to include or use extra information that is additional to users and items in order to determine user preferences on not known items. Contexts represent a set of explicit variables that model contextual factors such as time/date, places, equipment and incidents. The CARS process takes one of the following forms: Contextual pre-filtering, Contextual post-filtering and Contextual modelling (Adomavicius & Tuzhilin, 2015). In Contextual pre-filtering the information about the current context is used only for picking the associated set of data and then the ratings are estimated using any traditional RS on the chosen data. In Contextual post-filtering, the information concerning the current context is not used and the ratings are estimated using any traditional RS on the full data. The final list of recommendations is settled for each user using the contextual information. In Contextual modeling, the information is applied straight to the modeling procedure as part of the rating prediction (Adomavicius et al., 2011).

The work in (He et al., 2010) introduces a context-aware citation RS that calculates the context importance between a citation and a document in order to help users on finding the topic of a paper. The proposed method suggests high-quality citations for a context and shows the effectiveness and the scalability of the technique compared to other methods. Another personalized RS that uses articles is presented in (Li et al., 2010) to address the contextual bandit problem. A bandit problem is a method where a training algorithm chooses articles in a particular order to assist every user on choosing an article by applying the contextual information between users and articles. At the same time, it modifies its selection strategy based on a user-click feedback to raise the total user clicks. Results show an improvement on clicks compared to a regular technique that makes no use of context. Moreover, CF methods based on Tensor factorization models like the user-item-context on any dimension tensor rather a 2D user-item matrix outperforms the OLAP approach and Item splitting method, which are two well-known context-aware procedures (Karatzoglou et al., 2010).

The work in (Gavalas & Kenteris, 2011) introduces a mobile travel RS that makes use of collaborative techniques. The proposed method considers any contextual information in order to produce improved recommendations to eliminate the problem that existing travel RS are facing, which is the fact that they are not using the information, behaviors and evaluations of other similar persons. In this work a network using wireless sensors is used to enable correct localization and gives to users free mediums for transferring any information and ratings about interesting places using their mobile phones. Finally, this study presents a context-aware rating concept in which the user's ratings that are uploaded within the network premises have a higher weight than the ratings from other users that did not visit the location.

Moreover, a significant challenge in RS is to capture the user preferences over time with outstanding accuracy (Xiang et al., 2010). Users change their preferences over time due to not-known happenings. Users' behavior can often be defined by user's short-term and long-term preferences (Xiang et al., 2010). According to (Koren, 2010) consumer preferences for products are changing over time. Product attention and popularity are regularly fluctuating making customers reconsider their tastes, interests and feelings so when designing RS developers should take into an account temporal dynamics.

A Session-based Temporal Graph (STG) that models the users' long-term and short-term preferences over time is introduced in (Xiang et al., 2010). Based on the suggested structure, an innovative recommendation method that is called Injected Preference Fusion (IPF) is composed and the personalized Random Walk method for temporal recommendation is expanded. The proposed method provides notable improvements over other state-of-the-art algorithms, such as the user-based, the item-based and the Page rank including temporal data. Finally, the author in (Koren, 2010) outlines a model that tracks the time changing performance during the lifetime of data and uses two CF recommendation strategies (neighbourhood and factorization methods) to significantly improve the quality of predictions.

3.2 Tagging Systems

The Social tagging systems provide three distinct types of recommendations (Milicevic et al., 2010). Firstly, they suggest tags to users based on the tags that other users used on similar items. Secondly, they recommend items to users based on similar tags a user is interested compared to other users. Finally, they propose users with similar interests based on the same tags that they did on related items (Symeonidis et al., 2010).

The factorization models based on the Tucker Decomposition model provide high-quality tag recommendations exceeding other approaches like PageRank, FolkRank and CF methods (Rendle & Schmidt-Thieme, 2010). A factor analysis approach based on the probabilistic matrix factorization is stated in (Ma et al., 2011) to solve the sparsity problem by combining a social network and social tags. The proposed method performs much better than the state-of-art CF approaches of non-negative matrix factorization (NMF) and probabilistic matrix factorization (PMF). A different approach of tagging systems is introduced by the authors in (Symeonidis et al., 2010); this method models the data in a 3-order tensor in which the latent semantic analysis and the dimensionality reduction are produced using higher order SVD and Kernel-SVD techniques to solve the problem that similar users may have varied interests for an item. The suggested algorithm improves the recommendations in terms of effectiveness compared to other recommendation algorithms such as Item-based CF, Matrix SVD, Fusion, Folkrank and Collaborative Tag Suggestion. Moreover, a graph-based algorithm is outlined in (Guan et al., 2010) that finds the associations between users, tags and documents by presenting them in an equal semantic space to eliminate the problem of non-accurate document recommendation. The proposed algorithm exceeds traditional recommendation algorithms such as User-based CF, Sunk-SVD, Tag Vector Similarity and Rocchio CB.

The tagging recommendation algorithm proposed in (Rashid et al., 2002) is based on an integrated diffusion on the user-item-tag tripartite graph (Zhang et al., 2010) that tries to eliminate the challenges of accuracy, diversification and novelty when the data set is sparse. Results show that using the tag information the accuracy, diversification and novelty are significantly improved. Another approach that deals with sparsity is the one presented in (Kim et al., 2010). That work outlines a collaborative tagging approach to filter user's preferences for items to tackle aforementioned problem as well as the cold-start problem in order to provide a better recommendation quality. A tool that locates the notions of the tags is outlined in (Cantador et al., 2011); this tool maps the tags in semantic objects that belong to external knowledge bases by exploiting ontologies; ontologies help the tool to automatically filter and classify the exposed tags in a set of purpose-oriented categories in order to remove the belief that a percentage of the tags is noisy.

Additionally, two unique document- centered approaches make effective tag recommendations by grouping a recent document in one or more topic classes and then select the most relevant tags from those groups to automate the process of making tag recommendations to users when a new resource becomes available. The first method describes the tagged data in two bigraphs whose points can be partitioned into two sets that do not have elements in common and the second approach detects the most typical documents within a data set and uses a Gaussian process classifier for effective classification. The two approaches enhance the performance of tag recommendations when compared to user-centered methods and topic models such as Linear Discrimination (LDA) and Support Vector Machines (SVM) classifiers (Song et al., 2011). Furthermore, the approach described in (Kim et al., 2011) first identifies related and unrelated topics for users and then improves a particular user model with the collaboration of other similar users. Experimental results show that the proposed model provides a better representation of user interests when compared to a user-based, item-based and TF-IDF.

In addition, the personalized algorithm for recommendations in folksonomies proposed in (Shepitsen et al., 2008) relies on hierarchical tag clusters and reduces the cost of having a bad vocabulary that can result in tag redundancy. Using data mining techniques (Mobasher et al., 2002), such as clustering, the model provides means to correct the aforementioned problems by identifying trends and eliminating noisy data. Folksonomies containing only one topic than many topics perform an obvious target for suggestions since they are more concentrated and less rare. The Pairwise Interaction Tensor Factorization model (PITF) described in (Rendle & Schmidt-Thieme, 2010) is a unique case of the Tucker Decomposition model. It uses a continuous runtime for learning and prediction to eliminate the cubic core tensor problem of the Tucker Decomposition models. The PITF model is acquired using a variation of the Bayesian personalized ranking (BPR) criterion that was first proposed for item recommendations. The proposed method outperforms the Tucker Decomposition model in both runtime and prediction accuracy.

According to (Guy et al., 2010), a RS that gives to each suggested item a clear explanation which involves the people and tags used for the production of its recommendation and the connections between the users shows a greater acceptance ratio for a tag-based RS than for a people-based RS and even a sufficient performance for a combined system. Moreover, the integration of tag and time information when prognosticating users' preferences in CF provides suitably personalized recommendations for social tagging schemes (Zheng & Li, 2011).

Moreover, besides the classic tagging systems the work in (Cacheda et al., 2011) presents a fuzzy linguistic modeling (FLM) which is a mechanism based on language variables that provides significant results when modeling information for decision making, information retrieval and political analysis. FLM was introduced in RS in order to develop systems that can deal with the challenges of RS. The work in (Porcel & Herrera-Viedma, 2010) presents a new fuzzy linguistic RS that uses the acquisition of user preferences to identify user profiles. Users provide their preferences through a not complete fuzzy linguistic approach. The proposed system operates as a decision support system and makes choices about the resources that could be interesting for a researcher, or it recommends collaboration opportunities with other researchers targeting to develop interesting working groups. Furthermore, a fuzzy linguistic RS based on the capabilities of the Google Wave service is used as a tool for a better communication between the researchers interested in related research areas (Serrano-Guerrero et al., 2011). The recommendations are produced according to several pre-defined characteristics that utilize fuzzy linguistic labels. The system supports potential collaborations between multi-disciplinary researchers and recommends extra resources interested for interaction. The proposed methodology was tested from several research groups of the same university achieving successful results. Finally, the work in (Zhu et al., 2014) recommends a novel tagging approach, that aims to protect users under the notion of differential privacy. Experimental evaluation shows that the proposed algorithm can effectively retain the utility of the datasets keeping at the same time the privacy in high levels.

3.3 Mobile Recommender Systems

Nowadays people are able to log time and location data anywhere in the world by making use of various technologies like GPS, GSM, GIS etc. The capture of real-world histories that include the user's interests for specific locations offers many possibilities and can help us to better understand the relationship between users and interesting places (Zheng et al., 2011). The mobile RS use traditional CF systems based on explicit ratings to obtain the user preferences but they come up with some limitations (Chiu et al., 2010). Mobile customers find it tough to determine their tastes instantly because of poor interfaces and high expenses. Implicit ratings are more favorable for mobile RS but they usually utilize ratio scales for expressing preferences which are also undesirable as they may increase estimation errors (Lee et al., 2010).

The CF recommendation method introduced in (Lee et al., 2010) is based on implicit ratings and less ordinal scales, and tries to eliminate the problem that mobile users face who may find it difficult to rate their suggestions using explicit ratings. A mobile web usage mining approach is proposed in (Lee et al., 2010) to capture the implicit ratings and then a model used in decision-making is applied to form the consumer profiles. Results confirm that the proposed methodology produces better performance than existing CF algorithms. An innovative content service on mobile devices is introduced in (Liu et al., 2011) and is used to filter and promote blog articles to users that have an interest in them. The m-CCS system offers a novel approach that foretells the latest popular blog topics based on popular weblogs on a specific time. It summarizes the users' browsing logs to determine their interests which are then joined with the newest hot blog topics to determine their favored blog theme or articles. This hybrid approach recommends articles by combining the reputation of topic groups for each user, item-based CF and the number of clicks per article. Results show that the m-CCS system can definitely suggest interesting articles to mobile users.

The RS proposed by authors in (Miller et al., 2003) uses PDAs which are connected to the Internet to supports the users of a movie service on easily selecting movies to rent, buy and see when they are away of their computer. This system presents the ability of a mobile RS to produce essential value to their clients (Miller et al.2003). Moreover, the mobile-based system outlined in (Quercia et al., 2010) carries out a research specifying on user preferences interested in social events in different locations. The use of mobile RS can assist users by recommending interesting events to visit even if there are numerous social activities in a day. The best performing algorithm in this work suggests interesting events among the inhabitants of the same area with high accuracy.

The authors in (Zheng et al., 2011) propose a personalized friend and location RS that uses a Geographical Information System (GIS). The proposed RS captures a person's visits in a region as implicit rating and are then uses that information to measure the similarity between different users before producing recommendations to each user a group. In addition, the proposed system suggests a collection of not visited places that the individual may be interested in. Also the hierarchical-graph-based similarity metric (HGSM) introduced in this work is used to control each person's location history and compute the similarity among users. The proposed HGSM metric outperforms other similarity measures such as the Cosine and Pearson similarity and provides to users numerous interesting locations satisfying their overall experience. Finally, the study in (Mo et al., 2014) proposes a cloud-based mobile RS which can moderate network overhead speeding at the same time the recommendation process. The users are grouped into several clusters according to their context types, their information is captured from video-sharing websites, to produce multimedia recommendation rules based on the Hadoop platform. When a new user request comes, the rules are optimized to generate real-time recommendations. Experimental results present that the proposed methodology can recommend anticipated services with high precision, high recall, and low response delay.

3.4 Group Recommendations

The majority of RS makes recommendations for unique users, but in some cases the items that are selected for personal use are not suitable for a group; therefore, in order to produce efficient recommendations for a crowd, the RS must serve the personal preferences of all group members (Baltrunas et al., 2010).

The effectiveness of a group recommendation CF system is achieved by combining the unique recommendations lists of each user using a normalized discounted cumulative gain. This method shows that the effectiveness is not decreasing when the group size grows and also demonstrates that its successfulness increases when there are users in a group with similar interests (Baltrunas et al., 2010). Another outcome from this work is the fact that when individual recommendations are not sufficient a user could obtain better suggestions looking at group recommendations. Moreover, the authors in (Garcia et al., 2011) outline a RS for tourists that classifies the users based on their location, their interests and the places that they have already visited; this approach offers accurate suggestions for specific users or a group of users using aggregation techniques to capture individual personal recommendations.

The work of (Masthoff, 2004) discusses different strategies for joining together personal user models that can be accommodated by groups; some of them are motivated by the Social Choice Theory. The first experiment of this work investigates how people select a series of objects for a group and shows that humans care about fairness in order to withdraw personal sadness. The second experiment studies how satisfied people are when the objects are recomemnded from different strategies and not just one. The series generated by various strategies give pleasure to all members of a group but they also give more emphasis on showing the best-rated item to each user. A final outcome that is observed is the fact that the rankings of the items at the beginning and the end of a sequence are critical. Finally, the study in (Roy et al., 2014) studies the problem of enabling the flexibility of updating a user's preferences in group recommendation. Any member provides a set of preferences which, in addition to its past preferences and other members' preferences, are utilized for producing group recommendation. Evaluation of proposed approach on real world data-sets validates the findings of the anticipated work.

3.5 Recommender Systems Using Data Mining Techniques

Web usage mining techniques can be applied to CF in order to address some of the weaknesses of RS, including the dependence on user ratings, scalability issues and the bad performance on scattered data (Mobasher et al., 2002).

The authors in (Mobasher et al., 2000) provide several techniques in which the user preferences are automatically determined by using association rules to solve the problem of old profile data as user preferences change over time. Results show that Web usage mining techniques help to increase the efficiency, accuracy, scalability and adaptability of RS. Another technique based on Web usage mining and product taxonomy is applied in (Cho & Kim, 2004) to improve the recommendation quality and the system performance of the existing CF-RS to reduce the problems of sparsity and scalability that lead to poor recommendations. Results show that the proposed methodology provides accurate results and performance than other state-of-the-art CF methods.

RS can use different techniques based on clustering of user characteristics to identify aggregate profiles for web personalization (Mobasher et al., 2002). By making use of aggregate profiles, a useful personalization at the early stages of a user's visit in a website is produced based on the click stream information without knowing anything about that specific user. The work in (Liu & Shih, 2005) introduces a product recommendation methodology that links data mining techniques with decision-making for groups to evaluate the Customer Lifetime Value (CLV). At first, clustering techniques are used to group the clients of a service and then association rules procedures are performed to provide recommendations to each member of a group. Experimental results show that this approach outperforms the standard CF method of k-NN. Moreover, an attribute reduction-based mining method is outlined in (Jung, 2012) that efficiently identifies domain experts (long-tail groups) who play an influential role as information sources are used in the recommendation process to produce accurate results.

A methodology based on a mixture of data mining techniques, such as decision trees, web mining, product taxonomy and association rules is presented in (Cho et al., 2002) to produce quality recommendations. Also, a different data mining method used for the formulation of an innovative RS is reported in (Duan et al., 2011). The proposed system uses the associations that exist among the diagnoses and the outcomes of a nursing system to produce a sorted list of proposed care plans. Furthermore, a personalized RS designed to suggest new products to supermarket buyers based on association rules is reported in (Lawrence et al., 2001). Association rules are applied to define the relationships among the products and then a clustering technique is utilized to classify customers into groups based on similar spending histories before generating recommendations. Experimental results show an increase in the supermarket's revenue as many people are choosing products from the recommendation list. Moreover, the work in (Lin et al., 2002) proposes a new collaborative recommendation method created to mine association rules in order to deal with the problem that many of those rules are not associated with a certain user. In this work assosiation rules are mined only for a specific user, decreasing the time needed for mining in the whole dataset providing accurate recommendations. Experimental evaluation exhibited better performance compared to traditional correlation approaches. Finally, the study of (Amatriain et al., 2015) outlines the most important techniques that can be used for classification purposes such as the k-means clustering, association rules and others presenting at the same time various cases where they can be applied with success.

3.6 Multi-criteria and Multi-dimensional Recommender Systems

The integration of multiple criteria into the CF processes eliminates the problem present in the single criterion systems that they may produce recommendations that do not meet user needs (Ya & Zhao, 2012).

The work in (Hwang, 2010) proposes a Genetic Algorithm to determine the weight of each user toward each feature which uses those weights into the CF process to provide recommendations improving the performance of the RS (Hwang, 2010). In addition, the work in (Adomavicius & Kwon, 2007) present a similarity-based approach and an aggregation function-based method to link the multi-criteria rating information in a RS improving the recommendation accuracy when compared with other techniques while the paper of (Adomavicius & Kwon, 2015) provides a brief overview of the class of multi-criteria RS. Moreover, a multi-attribute collaborative algorithm is proposed in (Manouselis et al., 2010) that implements a resource learning service for a society of teachers in Europe. Final outcomes present that such systems should take into consideration the certain communities that will serve in order to provide accurate recommendations.

In addition, the work in (Kazienko, Musiał & Kajdanowicz, 2011) outlines a multi-dimensional (MD) social network that utilizes the data obtained from users behavior and their shared activities. Various object-based associations are classified into layers to match social or semantic relations between individuals aiming to form personalized suggestions that are adapted to the users' needs improving the recommendation accuracy of the system. Moreover, a multi-dimensional strategy that supports various dimensions, profiling information and hierarchical aggregation of recommendations and provides precise recommendations based on additional contextual information besides the common information that exists on users and items is described in (Adomavicius et al., 2005). The authors in this work propose a merged rating predicting method that recognizes the conditions where the MD approach outperforms the standard 2D approach and use the MD method in those conditions and the standard 2D approach elsewhere in order to provide reliable recommendations on time. Finally, the work in (Nilashi et al., 2015) suggests a hybrid approach for hotel recommendation that uses dimensionality reduction and various prediction techniques. The authors have established a multi-criteria CF system to enhance the predictive accuracy by a using Gaussian mixture model with an Expectation Maximization (EM) algorithm and an Adaptive Neuro-Fuzzy Inference System (ANFIS). Experimental evaluation present that the proposed hybrid model provides high accuracy for hotel recommendation.

3.7 Recommender Systems in Cloud Computing

Cloud computing uses remote servers connected to the Internet to store, manage and process data. Today, many companies and corporations see this as an opportunity to replace their existing IT infrastructure due to the benefits of cloud services, such as flexible computing and low costs (Rehman & Hussain, 2011).

The work in (Zhang et al., 2011) recommends a neighborhood-based method for a quality prediction of elements in a Cloud system. The proposed system, namely CloudPred, faces the crucial challenge of using all available user-side elements of the Cloud for evaluation and demands no further request of Cloud components from the system's architect. Experimental results inidcate that the suggested approach delivers greater QoS prediction accuracy than other systems. Moreover, authors in (Zheng et al., 2010) propose a QoS component ranking structure that is another solution to handle the challenge of components quality aiming at selecting parts for the Cloud from a collection of components with similar characteristics. This procedure requires no additional demands from the application designers and outperforms competing approaches. Since Cloud services are described using multiple criteria (costs, privacy, components, performance) the work in (Rehman & Hussain, 2011) proposes a multi-criteria Cloud service methodology for selecting Cloud services.

Additionally, the study in (Han et al., 2009) outlines a Cloud Service framework that uses a RS which helps a user to choose services from various Cloud providers that suit the users' requirements dealing with the limitation of not knowing what service to choose from when the number of Cloud services is increasing. The proposed system suggests a service that depends on the network QoS and Virtual Machine factors of the various Cloud providers. Experimental results reveal that the proposed model recommends a suitable mixture of Cloud services to users. Moreover, the work in (Zhang et al., 2012) presents a new approach that is called CloudRecommender also used to select Cloud-based services. The proposed approach matches users' application needs with Cloud service configurations and then catches those configurations in an ontology in order to generate an automatic Cloud service selection. Finally, the work in (Vera-del-Campo et al., 2014) suggests a RS that aims to protect members against legal attacks. The authors first analyze the properties of possible deniability and anonymity of the system's nodes and then they use that information to suggest items to the clients hiding any data about the recommended item.

3.8 Recommender Systems with Ontologies and Linked-Data

Ontology is a machine-readable construction of a certain domain and consists of entities, attributes and relationships. Relationships describe various information such as an employee's address or phone number, or activities such as editing a file or attending an event. Ontologies are referred to the structure and instance classifications that exist in a knowledge base (Middleton et al., 2004).

An innovative approach dealing with user profiling within a RS is examined in (Middleton et al., 2004) and works to settle the problem of recommending academic research papers online. Two experimental systems are used to create user profiles by observing the behavior and feedback of users. In addition, the research papers are grouped by applying ontological classes and the CF approaches are used to recommend papers that have been seen by similar users. Final results show that ontological approaches improve the user's profiling accuracy. Moreover, the work in (García-Crespo et al., 2011) presents a semantic hotel RS that is based on the client's experience on previously seen suggestions (García-Crespo et al., 2011). The system uses the client's experience point of view to implement fuzzy logic methods to associate the client with hotel features that are expressed by domain ontologies. Results show that the recommendations provided by the proposed system are on the same level as a domain expert.

RS can significantly improve the user's experience, but there are many barriers as regards data acquisition; therefore, it is really difficult for a new service to compete present recommendation services (Heitmann, 2010). The WWW is moving away from websites that use hyperlinks towards a cyberspace of linked-data. An enormous number of RDF data is published for free in datasets that are linked together to form a cloud using linked-data. Nowadays, there are thousands of RDF data available on the Internet, but there are not so many applications that utilize their inherent power. The author in (Heitmann, 2010) presents the development of a RS that utilizes linked-data to overcome the sparsity problem and the cold-start challenge of RS. The proposed system uses data from various sources and then makes recommendations using CF techniques. Moreover, the work in (Di Noia et al., 2012) presents how Linked-data can be utilised for building a RS that is based only on the information from the Web. The proposed CB system utilizes data from Linked Open Data datasets (DBpedia and Freebase) and suggests movies to users outperforming other approaches in terms of accuracy. Finally, the authors in (Meymandpour et al., 2015) present how Linked Open Data can be a reliable and rich source of content information that can support RS to deal with the fundamental problems of cold-start and limited content analysis.

3.9 Deep Learning Recommender Systems

Deep learning techniques have been applied with high success in a wide area of applications like image and speech recognition (Chung, 2015). According to the literature, a lot of researchers started using those techniques in RS to provide better recommendations solutions that deal with data sparsity and uncertainty.

The work in (Wang et al., 2015) presents a hierarchical Bayesian model called Collaborative Deep Learning (CDL) for RS to address the data sparsity problem. The proposed approach performs deep learning between the content information and the collaborative filtering ratings matrix. Experiments on real-world datasets from various domains demonstrate that the use of CDL advances the state-of-the-art methods. The study in (Salakhutdinov et al., 2004) claims that most of the existing collaborative filtering methods cannot deal with very large datasets and presents a class of two-layer undirected graphical models called Restricted Boltzmann Machines (RBM) that can be used to model tabular data. A set of efficient learning and inference approaches are presented for the RBM model and applied on the Netflix dataset. Evaluation results showed that the RBM approach outperformed the Singular Value Decomposition (SVD) models.

The authors in (Zhang et al., 2014) argue that click prediction is one of the main problems in sponsored search and that most works in the literature make use of machine learning approaches in order to predict ad-click for each event. In a real-world system users' behavior on advertisements depends on how they behaved in the past. That work introduced a novel framework based on a Recurrent Neural Network (RNN) network. The proposed RNN was evaluated using the click-through logs of a large scale commercial engine and the results indicated significant improvements on the click prediction accuracy compared to sequence-independent approaches.

3.10 Session-based recommendations

Session-based recommendation is a recent challenge in the area of RS introduced in the RecSys Challenge of 2015 (Hidasi et al., 2015). In this context a RS provides recommendations taking into consideration only the actions of users in a current browsing session (Tan et al., 2016). This type of recommendations processes the historical data of users captured during an active session and relying only on a narrow piece of information that describes the behavior of a specific user it predicts the user's next move and recommends an item.

The success of deep neural networks when dealing with image or speech recognition (Russakovsky et al., 2015) was the initial starting point for incorporating such models in RS so as to deal with unstructured data and produce accurate session-based recommendations. The research in (Hidasi et al., 2015) outlines a RNN network applied on a challenge that most real-world RS face, that is, how to deal with long session-based data that exist on large ecommerce websites, and hence, with data sparsity. In this kind of problems, the frequently used matrix factorization approaches are not accurate enough so modeling the whole session can result in more accurate recommendations to users. The proposed methodology introduces several modifications to classic RNNs, such as the Gated Recurrent Unit (GRU) and a ranking loss function, but at the same time it considers practical aspects of the task. Evaluation is performed on two datasets, the first being that of the RecSys Challenge 2015 and the second one was collected from the OTT video service platform. Experimental results showed that the suggested approach outperforms the best baseline method of item-KNN on such problems and that further improvements can be performed when adjusting the RNN parameters and changing the loss function. Finally, the work in (Tan et al., 2016) further analyzes RNN-based models for session-based RS and proposes two techniques that lead to improvements on the models' performance. The work was evaluated using the dataset introduced in the RecSys Challenge 2015 and the results were compared against the models presented in (Hidasi et al., 2015).

3.11 Classification Models in RS

Machine learning techniques such as the Classification models of Support Vector Machines, Weighted k-NN, Linear Discrimination and Decision Trees etc. can be used as solutions in order to enhance the performance of RS by providing accurate predictions that can be used as recommendations. The work in (Candanedo & Feldheim, 2016) uses a Linear Discriminator Analysis (LDA), Regressions Trees and Random Forests for training and testing purposes in order to predict whether a room is occupied or not. Results showed that the impact of accuracy on each experiment depends on the classification model and the number of features selected each time. Taking into consideration all features the best accuracy was resulted using the LDA model for both test datasets.

Moreover, the authors in (Šter & Dobnikar, 1996) present a number of classification systems on various medical datasets (Diabetes, Breast Cancer and Hepatitis) in order to obtain accurate results when utilizing a number of various machine learning methods. In terms of classification accuracy in most of the datasets the neural networks approaches outperform other methods such as Linear Discrimination Analysis (LDA), K-nearest neighbor, Decision Trees and Naïve Bayes. In addition, the study of (Shin et al., 2005) investigates the application of a SVM model to a bankruptcy prediction problem. Even though it is known from previous studies that the back-propagation neural network (BPN) produces accurate results when dealing with pattern recognition tasks, it faces limitations on constructing an appropriate model for real-time predictions. The proposed classification model based on SVM captures the characteristics of a feature space and is able to find optimal solutions using small sets of data. The suggested approach performs better than the BPN in terms of accuracy and performance when the training size decreases.

3.12 User Privacy & Trust in Recommender Systems

Privacy risk is an essential challenge of RS that needs to be addressed. Computing provides huge potential for persons to share all kind of information regarding their locations or preferences, but the privacy risks are severe (Canny, 2002). Shoppers do not trust inaccurate recommendations from a RS that has a restricted or limited database. E-commerce sites with limited databases have to merge their databases for two main reasons: 1) to enhance the genuineness of the recommendations and 2) to maximize the accuracy of targeted audience protecting at the same time the privacy of users (Zhan et al., 2010).

The author in (Canny, 2002) presents a new probabilistic factor analysis model that protects the privacy of users' data. The privacy protection is implemented by a p2p protocol and the proposed approach handles the missing data without expecting default values for them. The presented model is one of the most accurate methods for protecting users' privacy in RS and also has additional benefits as regards to the speed and storage of the RS compared to other approaches such as the Pearson correlation and SVD. Moreover, the work in (Zhan et al., 2010) addresses how to bypass privacy exposure in CF systems by comparing different cryptography strategies and forming an efficient privacy protective scheme based on the scalar product protocol without exposing customers' private data. In addition, the authors in (Harman et al., 2014) present two psychological experiments (N=400) to evaluate trust in RS over time, under personalized and non-personalized recommendations. The final outcomes show that Humans trust inaccurate recommendations more than they should.

Furthermore, a trust-aware RS model determines that a trust network is a system in which the distance between two randomly chosen nodes is small (Yuan et al., 2010). The work in (Yuan et al., 2010) confirms that by involving people in the recommendation process that are close with the active users gives you high rating coverage reducing at the same time the predicted rating error. Moreover, the trust recommendation model presented in (Walter et al., 2008) operates on a social network and shows that agents use their social network in order to transfer information. Results show that the proposed RS organizes itself in a state with a tremendous performance that is obtained without the explicit coordination from the local interactions of the representatives. This model was the start of developing complex models of RS by linking together the theories of social networking and trust relationships. A method outlined by the authors in (Wei et al., 2011) deals with no explicit ratings by transforming a social recommendation model. The network's topology is improved by the proposed algorithms in terms of performance as any other not-scalable method that use extra information. The metric proposed in (Massa & Avesani, 2007) presents the trust over a network and calculates a trust weight that is used to replace the similarity weight in RS to address the Sparsity problem. Evaluation results show that using the proposed metric is more useful than any other CF measure when implementing a trust-aware RS. Finally, according to the literature, more automated collaborative systems should be produced since current RS are like black boxes that provide no clarity and explanations back to the users. Users will trust a system when they know the reasons that generated a suggestion (Herlocker et al., 2000).

3.13 Shilling Recommendations

RS are extremely exposed to shilling attacks, both by persons and groups. Violators submit biased ratings to change the recommendations of a RS in order to influence negatively its algorithms (Zhou et al., 2015). The work presented in (Lam & Riedl, 2004) proves that unethical producers find it beneficial to shill RS by posting false reviews in order to have their goods recommended more frequently than those of their opponents.

The authors in (Chirita et al., 2005) outline that CF techniques prove to be more exposed to attacks than other methods as infected user profiles can insert into the RS in order to promote specific items. A number of open questions that may affect the effectiveness of shilling attacks are explored in (Lam & Riedl, 2004) and include the following: Which algorithms are being used by the system, if the application is generating recommendations, how an operator can detect these attacks and what are the characteristics of the objects that are being attacked. Results show that the item-based method was less influenced by the attacks when compared with the user-based approach. Most specifically, most attacks reported to date consider a small amount of knowledge regarding the RS and target algorithms like the k-NN and user-based. Experimental results advise new ways that can be used to judge and detect shilling attacks on RS.

Additionally, the authors in (Zhou et al., 2015) utilize statistical metrics for identifying the rating models of attackers and to group features that exist in attacked profiles that were ignored in previous studies. The Rating Deviation from Mean Agreement measure (RDMA) and the Degree of Similarity with Top Neighbors measure (DegSim) are used to analyze a rating pattern and separate an infected profile from a true one. Outcomes show that the detection model introduced in (Zhou et al., 2015) based on a target item analysis detects and deals with shilling attacks more effectively than any other method. Finally, the work in (Chirita et al., 2005) presents various metrics for defining the rating patterns of infected profiles of users and implements an approach that is used to detect any shilling attack into the RS. The proposed methodology monitors the ratings of a user continuously removing any malicious profiles from the system and then uses the remaining non-infected profiles to compute the recommendation list achieving high quality resulting.

3.14 Metrics

According to the literature there are several metrics that can be grouped into categories which are used to measure the accuracy and performance of the RS in order to evaluate them correctly (Pu et al., 2012).

The first category comprises error metrics and includes the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE). The MAE metric is computed using all ratings available in the evaluation set and measures the difference between the prediction and the real rating using the absolute value (Lü et al., 2012). Moreover, the RMSE metric is providing a higher emphasis on larger errors and measures the differences between the predicted values and actual values.

The second main category considers various evaluation metrics that are used to separate good from bad recommendations (Lü et al., 2012) like precision and recall. Precision measure is defined as the ratio of relevant items to recommended items and recall is the relationship of all essential objects that are recommended to the total number of related items. When the precision increases, the recall decreases so in order to consider both metrics under one calculation the F1 metric introduced in (Pu et al., 2012) can be used. In addition, the coverage metric which is the percentage of the items a RS can suggest is used to identify the algorithms that despite their sufficient accuracy they recommend only a small number of objects which are popular items that the user is already familiar with (Lü et al., 2012). Finally, various metrics have been proposed to evaluate the usefulness of an ordered list of recommended items. One of them is the half-life utility metric which is based on the experience that the items that appear at the beginning of a list have a higher possibility of being noticed and chosen by users than the others that follow (Konstan & Riedl, 2012).

3.14.1 Diversity and Novelty

Except from the evaluation and error metrics there are also measures that go beyond accuracy, like diversity, novelty and serendipity (Herlocker et al., 2004). As the use of RS on the World Wide Web (WWW) increases the need to develop additional evaluation frameworks for recommendation approaches arises. The new categories of metrics must not only examine the accuracy and performance of the recommendation results but to be used for other purposes (Bobadilla et al., 2011).

A framework for the definition of the novelty and diversity metrics is introduced by the authors in (Vargas & Castells, 2011). That work examines these metrics using a combination of the choice, discovery, relevance and the principles of a probabilistic model. Experimental results demonstrated the properties of the proposed metrics. Moreover, the authors in (Zhoua et al., 2010) describe a new algorithm that addresses the challenge that most accurate results are achieved by methods that suggest items based on user or item similarities. The diversity metric used in a hybrid RS with an accuracy-focused algorithm is applied to solve the aforementioned challenge. Experimental results show that by implementing a hybrid system, more advantages can be achieved in diversity and accuracy without depending on any particular information.

The work in (Lathia et al., 2010) studies the temporal characteristics of a system as current evaluations do not take into consideration the fact that users resume rating items over time. Different CF methods such as kNN and SVD reveal that the size and time between the ratings influence diversity. Additionally, the item ranking techniques explored in (Adomavicius & Kwon, 2012) can generate more useful recommendations throughout all users of a RS helping at the same time the system's accuracy. Finally, the work in (Bobadilla et al., 2011) proposes other metrics to evaluate the novelty of users' recommended lists and the trust in their neighbourhoods.

3.14.2 Other RS Evaluation Metrics

Many discussions on why the recommender research community should move beyond accuracy to create a new user direction for the evaluation of RS are presented in (McNee et al., 2006). Researchers need to create a mixture of metrics that will not only act on the items that appear on a list but also on the list itself.

The lack of ratings offers valuable information for increasing the top-k hit rate. The author in (Steck, 2010) presents a performance measure that is computed using data even when ratings are missing (MNAR). Depending on the value of *k,* the proposed approach results in a higher hit rate as opposed to other state-of-the-art RS. An additional metric that combines the scientific information of the votes with some other autonomous information is presented in (Bobadilla et al., 2010). This metric uses the Jaccard measure and the Mean Square Error Metric to improve the results of Pearson correlation and operates only with the data that are stored by the RS users.

A new metric of user similarity is outlined in (Shang et al., 2010) and isbased on the user's preferences and the tagging information. The proposed measure is performing better than the cosine similarity metric because the similarities between the users are now estimated from a diffusion-based process. Additionally, the new metric presented by the authors in (Bobadilla, Ortega, Hernando & Alcalá, 2011) measures the similarity between users in a CF procedure using a linear mixture of values and weights. The values are now measured for each combination of users and the weights are now calculated using a genetic algorithm that operates on the data for each RS resulting in significant improvements on the quality and performance.

A metric that is called tendencies-based with SVD is presented in (Cacheda et al., 2011) which performs well under Sparsity conditions. Finally, the work in (Ge et al., 2010) concentrates on two critical measures used to evaluate a RS, the coverage and the serendipity. Results show that using different ways for measuring these metrics can lead to an improved user satisfaction.

3.15 Overview

A lot of research was conducted all these years focusing on solving the main limitations of RS; thus, now researchers need to concentrate more on the possible extensions of the capabilities of such systems described in this thesis. The aforementioned topics are only a few of possible enhancements of RS that can be used to improve the user's overall experience and the recommendations' accuracy, quality and performance and even to make the use of RS more capable in a larger spectrum of applications (Adomavicius & Tuzhilin, 2005).

The following sections present the development of novel RS systems that deal with the fundamental problem of RS aiming at the same time to provide accurate recommendations back to users. The research presented in this work was divided into three core parts which are presented in the sections below with details.

Chapter 4: Introducing Intelligent Multi-Criteria Systems

4.1 Introduction

This section presents the methodologies used for the development of multi-criteria RS that aim to tackle the challenges of RS and at the same time to improve accuracy when compared with other techniques.

We firstly intoduce a RS that captures the user's actions in a real-world environment and produces movie recommendations based on users' preferences which are dynamically changing over time. Moreover, we present a framework for deploying a RS in a shop environment that targets on recommending real-time personalized offers to customers by taking into consideration users' static and dynamic information. Final results indicate that the proposed methodologies outperform other state-of-the-art approaches and produce accurate recommendations related to what a user is interested in.

4.2 Technical background

4.2.1 Mapping the Data

First, we developed a preprocessing tool able to deal and convert any dataset to a specific format readable by our algorithms. Due to the fact that most datasets have different formats and structures, the need for such a tool was imperative. By utilizing the proposed tool any string dataset can be converted into a numeric one reducing its size and at the same time making the computational process more efficient.

To start with, the preprocessing tool consists of rules that find the dataset's size and the type of data for any specific column. Table 1 presents a set of rules that are applied on different types of data items in order to convert them to numeric ones. The first rule is applied on any numeric data and inserts the data into a new table, while the second rule converts the columns that consist of string characters into numeric values and then inserts them in the new numeric table. The third rule converts the dates into numeric values and moves them into the new table, the smallest date d_{min} is mapped to the value of 1 and every other date d is mapped to 1+`days_between`(d_{min}, d). The final rule converts big integers into a range between 0 and 1 using the standard normalization formula $\frac{n}{n_{max}-n_{min}}$ and also inserts the new data into the new table.

Table 1: List of rules for mapping a dataset

Rule	Data	Conversion
Move numeric columns into a new array	User ID, Others IDs, Numeric values	--
Convert string columns into numeric values	a, b, c, d etc. Word1, Word2, Word3 etc.	1,2,3,4 etc.
Convert Dates into numeric values	01/01/2016, 02/01/2016 etc.	1,2 etc.
Convert big integers into numeric scale	6903928, 3029102 etc.	0-1
….	….	….

The aforementioned rules can easily be extended to include other specific types of data. When all rules are finalized we come up with a new numeric dataset able to be used by the methodologies described in the next sections.

4.2.2 Determining the Number of Clusters

In this section we present the Entropy-based algorithm (Stylianou & Andreou, 2007) that executes when converting the dataset into numeric values. The Entropy-based approach is an external cluster evaluation measure (Stylianou & Andreou, 2007) that groups data objects with similar characteristics into clusters based on the entropy values of the objects using a similarity measure. For the purposes of the multi-Criteria RS framework, this methodology was used to compute the number of clusters that exist within a dataset, as well as their centroids.

Below we described how the Entropy-based method works:

The entropy value H_{ij} between two data objects X_i and X_j is defined as follows in equation (1),

$$H_{ij} = E_{ij}log_2(E_{ij}) - (1 - E_{ij})log_2(1 - E_{ij}) \quad (1)$$

where i ≠ j.

E_{ij} is the similarity measure between objects X_i and X_j and it is measured as shown in equation (2),

$$E_{ij} = e^{-aD_{ij}} \tag{2}$$

Where D_{ij} is the distance between X_i and X_j.

a is calculated using equation (3)

$$a = \frac{-\ln{(0.5)}}{\overline{D}} \tag{3}$$

where $\overline{D}$ is the mean distance among all data objects.

The total entropy value of X_i with respect to all other data objects is computed with the following equation (4),

$$H_i = -\sum_{\substack{j=1 \\ i \neq k}}^{n} [E_{ij} log_2(E_{ij}) - (1 - E_{ij}) log_2(1 - E_{ij})] \tag{4}$$

where i ≠ j and $E_{ij} \neq 1$.

In more detail the algorithm consists of the following steps:

1. Select a threshold of similarity β and set the initial number of clusters to $k=0$.

2. Determine the total entropy values H for each data object X.

3. Set $k = k+1$.

4. Select the data object X_{min} with the least entropy H_{min} and set $Z_c = X_{min}$ as the c_{th} cluster center.

5. Remove X_{min} and all data objects having similarity $\leq \beta$

6. If *dataset* is empty then terminate, otherwise go to step 3.

As already mentioned the Entropy-based approach is executed in the numeric dataset to determine the number of clusters and their centers. In the following we describe how we can group similar users using various clustering techniques.

4.2.3 Clustering techniques

4.2.3.1 Hard K-Modes

For the purposes of our framework and to deal with potentially large datasets, we choose Hard K-Modes as a suitable clustering technique for grouping each user or item on specific clusters in relation to the centroids.

The Hard K-Modes clustering algorithm (Huang, 1998) clusters categorical data by removing the numeric-only limitation imposed by other clustering techniques (e.g., k-means), using a matching dissimilarity measure. This feature of the Hard k-modes algorithm enables its efficient use for clustering large categorical datasets.

According to (Huang, 1998) there are two basic modifications between the k-means and the k-modes algorithm. Firstly, the Euclidean distance used in the k-means algorithm to calculate the distance between two objects is replaced by a dissimilarity measure such as Hamming, Cosine, Jaccard etc. Secondly, in the k-modes algorithm the cluster centers are represented by vectors of modes of categorical attributes. The mode of a set of values is the most frequent occurring value and there can be more than one mode in a set of values (Zhexue & Ng, 1999) as modes are updated with the most frequent categorical values in iteration of the clustering process.

The Hard K-modes approach is analyzed as follows:

Let X_1 and X_2 be two data objects of X defined by m attributes. The dissimilarity between the two objects is stated as in equation (5):

$$d(X_1, X_2) = \sum_{j=1}^{m} \delta(x_{1j}, x_{2j}) \tag{5}$$

where δ is described by,

$$\delta(x_{1j}, x_{2j}) = \begin{cases} 0, x_{1j,} = x_{2j} \\ \\ 1, x_{1j} \neq x_{2j} \end{cases} \tag{6}$$

If an object X_i in a given iteration has the shortest distance from a cluster center Z_l, it is represented by setting the value of the nearest cluster equal to 1 and the values of the other clusters to 0.

The objective function of Hard K-modes is presented as follows,

$$F(W,Z) = \sum_{l=1}^{k} \sum_{i=1}^{n} w_{li}^{a} d(Z_l, X_i) \tag{7}$$

where α is the fuzziness exponent, w_{li} is the weight degree of an object belonging to a cluster and k is the number of clusters. Weights are organized as,

$$W = [w_{li}] \text{ is a } k\ x\ n\ matrix \tag{8}$$

while equation (9) provides the set description for cluster centers,

$$Z = [Z_1, .., Z_k] \varepsilon R^{mk} \tag{9}$$

With hard clustering $\alpha = 1$ and the weight degree w_{li} of an object belonging to a cluster is given in equation (10).

$$w_{li} = \begin{cases} 1, if d(Z_l, X_i) \leq \ d(Z_h, X_i), 1 \leq h \leq k \\ \\ 0, otherwise \end{cases} \tag{10}$$

More specifically, the Hard K-modes algorithm consists of the following steps:

1. Select K initial modes, one for each cluster;

2. Allocate a data object to a cluster whose mode is nearest to the selected one;

3. Compute the new modes for all clusters;

4. Repeat steps 2 and 3 until no data object has changed the cluster membership.

4.2.3.2 Fuzzy K-modes

Besides the Hard K-Modes, (Zhexue & Ng, 1999) presents an extension of the Hard K-Modes algorithm namely, Fuzzy K-Modes. This algorithm was introduced in order to incorporate the idea of fuzziness and uncertainty in datasets.

The idea behind this variation is presented in equation 11. If an object shares the same values with a cluster, then it will be assigned entirely to that cluster and not to the others. If a data object in not completely identical to a cluster (*a > 1*), then it will be assigned to each cluster with a membership degree.

For *a >1*:

$$w_{li} = \begin{cases} 1, X_i = Z_l \\ 0, X_i = Z_l, h \neq 1 \\ \dfrac{1}{\sum_{h=1}^{k} \left[\dfrac{d(Z_l, X_{i,})}{d(Z_{h,} X_{i,})}\right]^{1/a-1}}, X_i \neq Z_l, X_i \neq Z_h, 1 \leq h \leq k \end{cases} \tag{11}$$

The Fuzzy K-modes algorithm consists of the following steps:

1. Specify the value of the weighting exponent;

2. Measure the proximity of components (which is calculated by simply matching the attributes of the component to those of the cluster centres);

3. Update cluster centres accordingly. The new centres are found by computing the mode from the categories of attributes that achieve the highest summation of membership degrees in the cluster.

4. Repeat steps 2 and 3 until the algorithm converges.

The end-result is a partition matrix holding the membership degrees of components to clusters, as well as the finalised cluster centres.

4.3 Multi-Criteria Movie Recommender Systems

In this section we introduce a new form of RS that utilizes multiple criteria in order to produce accurate recommendations to users. This type of RS was implemented and applied on the popular MovieLens dataset and was evaluated using the RMSE error metric. Experimental results showed that the proposed model outperforms other baselines methods. The proposed framework differs from the ordinary *users x items x ratings* RS as it does not require any ratings of items in order to provide recommendations to users; it just needs to learn the user preferences during a learning session. The proposed methodology utilizes the Entropy-based approach as well as the clustering techniques presented in the Technical Bakcground section. The methodology presented in here deals with the cold start problem as the clusters are formed based exclusively on users' preferences. Moreover, the information captured during the learning session is combined with the search input provided by the user in real time in order to create the clusters overcoming the data sparsity problem as even the most popular items will now belong to a cluster. Finally, the proposed approach provides recommendations quickly dealing with scalability issues as the recommendations are computed taking into consideration the cluster a user belongs to.

4.3.1 Related Work

The huge volume of information available on the Internet requires intelligent tools to search, retrieve and filter data so as to assist users in their everyday activities. RS have become powerful tools that serve this purpose by providing accurate recommendations to users, while additionally promote e-commerce and advertisement of goods over the Web (Jan de Nooij, 2008). This section provides a brief overview of works that deal with Multi-criteria systems.

The work in (Palanivel & Siavkumar, 2010) shows that Multi-criteria based systems can compute accurate recommendations by maintaining the details of user preferences in multiple aspects. Fuzzy sets seem to be an appropriate paradigm to effectively model the natural complexity of human behavior and to handle the fuzziness and uncertainty of human decision making behavior. Acknowledging the aforementioned benefits, that work adopts the fuzzy linguistic approach to efficiently present a Fuzzy Multi-Criteria Decision Making (FMCDM) approach to accurately rank the relevant items to a user. A Music Recommender System was developed to evaluate the performance of the proposed model, which was compared against the traditional item-based and user-based recommendation algorithms. Evaluation results showed that the proposed approach outperformed the aforementioned methods.

The authors in (Cheinshung & Chang, 2012) argue that RS that use the collaborative filtering technique and utilize single user ratings provide no useful information on users' preferences. When the number of existing users and items grows tremendously, traditional collaborative filtering algorithms suffer from serious scalability problems. To minimize the effects on scalability issues the authors proposed a cluster-based multi-criteria RS which integrates the genetic k-means algorithm into the collaborative filtering procedure. The genetic k-means algorithm finds the best possible partition of user clusters and uses those clusters to provide recommendations. Experimental results presented that the proposed methodology performed better in terms of accuracy and efficiency than classic collaborative filtering algorithms.

The work outlined in (Speigel et al., 2009) discusses the combination of collaborative and content-based filtering techniques in a neighbor-based prediction algorithm used in a web based RS. The performance of the proposed model was evaluated using the MoviesLens100K dataset that consists of 100000 ratings, 1682 movies and 943 users and was linked with IMDB.com to retrieve more content information about the movies. Final results outlined that the prediction accuracy of the proposed model was strongly dependent on the number of neighbors taken into account and that the item-based implementation produces better recommendations results quicker than the user-based.

The authors in (Ekstrand & Riedl, 2012) present a prediction analysis made by several well-known algorithms on the MovieLens10M dataset, which contains 10 million ratings and 100000 tag applications applied to 10000 movies by 72000 users. The users were divided into 5 sets and for each user 20% of her/his ratings were selected in each partition to be the test ratings for the dataset. Five recommender algorithms were then run on the dataset and the predictions of each algorithm for each test rating were captured. The results showed that the item-item algorithm achieves the highest accuracy compared to the other algorithms and also that for many cases in which one algorithm fails, there is another that will correctly predict the rating.

The work described in (Jung &Hay Pham, 2011) presents the need of long-tail users who can play an important role as information sources for providing accurate recommendations to short-head users. A case study on MovieLens dataset shows that 17.8% of the users can form a long-tail group. In order to evaluate the performance of the proposed system, 20 graduated students were invited to provide recommendations to others and give back their feedbacks. Following the proposed methodology 8 users out of 20 were selected as long-tail users (LTuG); this is two times higher than the MovieLens case study. Finally, it was also concluded that the user ratings of the LTuG could be used to provide relevant recommendations to the short-head users.

The work described in (Tsoukiàs et al, 2011) analyzes a hybrid framework that incorporates techniques from the field of Multiple-Criteria Decision Analysis, combined with a Collaborative Filtering approach. The proposed methodology improves the performance of simple Multi-rating RS for two main reasons, (i) the groups of user profiles are created before the application of the Collaborative Filtering algorithm, and (ii) these profiles are the result of a user modeling procedure, which is based on individual user's value system and exploits Multiple-criteria Decision Analysis techniques.

Adomavicius & YoungOk, (2011) propose two new approaches to take full advantage of the Multi-criteria ratings in various applications: (i) a similarity-based approach and (ii) an aggregation function-based approach. Both methods are used to incorporate and leverage multi-criteria rating information in RS. Multiple variations of the proposed approaches are discussed and an empirical analysis using a real-world dataset is performed. The experimental results show that multi-criteria ratings can successfully improve the recommendation accuracy compared to the traditional single-rating recommendation techniques.

The work in (Esparza et al., 2011) describes the benefits of Social Web that can be utilized by RS. Some information, such as tags, tweets, comments, likes, can be used as useful sources of user preferences and item information. The authors analyze a User-Generated Content (UGC) approach that is implemented for recommendations using various metrics such as coverage, novelty and diversity. The proposed method demonstrates superior performance when compared to user-based and item-based collaborative filtering techniques.

The work in (Luo & Zhao, 2012) outlines that by applying Multi-criteria techniques help us overcome the data sparsity and cold start problems that exist in single-criteria recommendation algorithms. The Multi-criteria recommendation algorithms proposed in that paper were used to perform prediction on two scenarios and to evaluate the customer's similarity. Experimental results show that Multi-criteria recommendation algorithms have no cold start problems due to the fact that when there is no customer record in the system, they recommend items using products' similarity. The proposed method also solves the data sparsity problem: in some cases, the customer evaluation may not be given thus causing problems of sparse data; in such circumstances, the multi-criteria recommendation algorithms compute the average value of customer evaluations to make it converge.

The work in (Manouselis & Costopoulou, 2007) explains a set of dimensions that distinguish, describe and categorize multi-criteria RS based on existing taxonomies and categorizations. These proportions are integrated into an overall framework used for the classification and analysis of existing multi-criteria RS. Final results provided an overview of the ways that current Multi-RS can support the decision of online users.

The work in (McNee et al, 2006) argues that RS do not always generate good recommendations back to the users and presents a RS to improve the quality of the recommendations by using a deeper understanding of users and their information. Human-Recommender Interaction (HRI) is a methodology that examines the recommendation process from an end-user's perspective and was used for analyzing user tasks and algorithms. HRI consists of three pillars: The Recommendation Dialog, the Recommender Personality and the User Information seeking Tasks that can lead to useful recommendation lists.

The work in (Karypis, 2001) presents a class of item-based recommendation algorithms that determine the similarities between various items to identify the set of items that can be recommended. Two methods were used in this work. The first method models items as vectors in the user space and uses the cosine function to measure the similarity between the items. The second method combines these similarities in order to compute a similarity between a basket of items and a recommender item. Five datasets were used for experimental purposes and the results showed that the effect of similarity for the cosine-based scheme improved from 0% to 6.5% and for the conditional probability from 3% to 12%. The effect of row normalization showed an improvement of 2.6% for the cosine-based method and 4.2% for the probability. The model size sensitivity test presented that the overall recommendations' accuracy of the item-based algorithms did not improve as the value of *k* was increased. Finally, it was concluded that the top-N recommendation algorithm improved the recommendations produced by the user-based algorithms up to 27% in terms of accuracy and at the same time was 28 times faster.

The work described in (Nadi et al., 2011) applied a hybrid collaboration and content based technique for developing a RS. The user receives accurate recommendations once the model analyzes the behavior of other users with similar patterns of interests. Evaluation results showed that using more efficient algorithms to find users with similar preferences leads to better RS producing at the same time more interesting recommendations.

(Alexandridis et al., 2013) addressed a problem that users tend to consume and rate items that are not similar to one another due to the fact that human taste or judgment is influenced by many factors that cannot be captured using content based or collaborative filtering. To overcome the aforementioned problem, a socially-aware personalized item clustering recommendation algorithm was proposed aiming at locating patterns between the items that a user liked by grouping them into different clusters. After clustering, members from each cluster were used to construct an item consumption network. At the end, by performing a walk on the network, accurate recommendations were produced that were also novel and diverse.

4.3.2 Dynamic Web Recommender System

4.3.2.1 Introduction

This part describes our first attempt to design a new dynamic Web Recommender System. The proposed system produces recommendations based on the preferences of the interested user, which are dynamically changed taking into account previous searches in real-time. This approach is enhanced by the utilization of static preferences which are declared by the user when registering into the system. The clustering procedure, being the heart of the recommendation engine, is of particular importance, and a number of techniques such as Entropy-Based, Hard K-modes and Fuzzy K-modes have been utilized. The proposed methodology was tested using the MovieLens1M dataset, which was linked with IMDB.com to retrieve more content information. The final results indicate that the proposed system meets the design objectives as it delivers items which are closely related to what the user would have liked to receive based on how he/she ranked the different categories depending on what he/she likes more and her/his previous behavior.

4.3.2.2 Methodology and Experimental Results

4.3.2.2.1 Proposed System

Figure 1 shows a schematic representation of the proposed system. First, the user registers into the system and ranks the different categories depending on what he/she likes more using a weight ranking system. The ranking of the categories is treated in this part as a user's static information because this type of information only changes after a certain period of time. Occasionally, the system asks the user to update her/his rankings due to the fact that interest in specific categories may have changed. Moreover, the system requires a certain number of searches to be conducted before start recommending items so as to understand the user's behaviour (dynamic information). A dynamic bit-string is created after the first searches and is updated after every new search, this string is compared with each movie in the dataset to eliminate those movies that the user is not interested in depending on the search profile thus far. The system then creates the clusters depending on the new dataset size (i.e. the movies in the lookup table) and the entropy threshold similarity value β which is assumed to be constant; however, its value needs to be tuned based on the size of the dataset in order to reach optimal performance. The next step is to update the clusters to include the static information of the user. This is performed so as to eliminate the problem encountered by the system after having a specific object belonging to two or more clusters. Therefore, new clusters also include the static information depending on how the user ranks the categories.

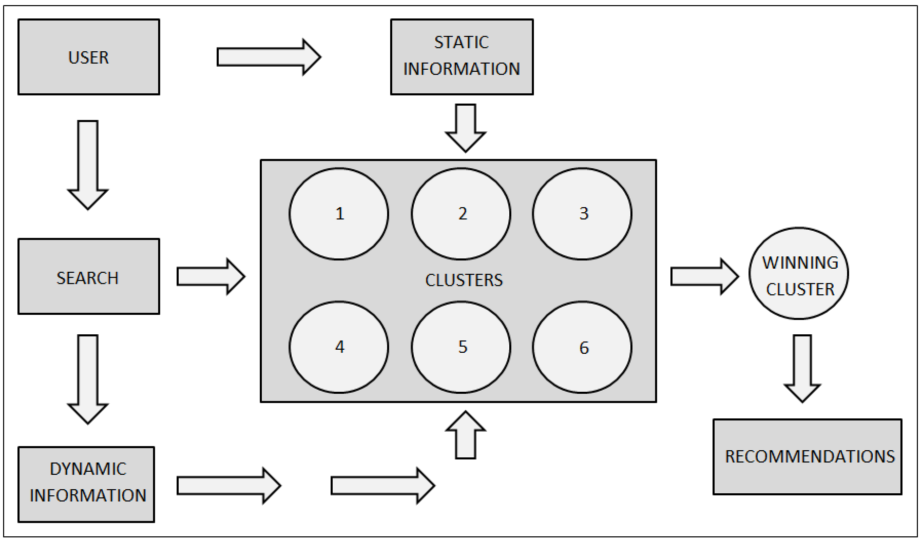

Figure 1: How the system works.

When a user performs single keyword queries the keyword used as search input is compared with each cluster center and the system finds the most similar cluster to the searched item (winning cluster); that cluster is then used to provide the recommendations. In the meantime, the searched keyword is saved as part of the dynamic information which is thus updated in real-time.

4.3.2.2.2 Dataset

The dataset used to evaluate the proposed system is an extension of the Movie Lens 1M dataset that can be found at GroupLens.org[1]. The Movie Lens 1M dataset consists of 1 million ratings from 6000 users on 4000 movies. The proposed method is not utilizing any user ratings so we only deal with the movies. From the 4000 movies some were duplicated and were removed; thus we concluded with a final dataset numbering a total of 3883 movies. We then linked the final dataset with IMDB.com, the world's largest movie database, to retrieve more information regarding the categories of each movie.

Table 2: Movie Categories

Column	Movie Category
1	Animation
2	Children
3	Comedy
4	Adventure
5	Fantasy
6	Romance
7	Drama
8	Action
9	Crime
10	Thriller
11	Horror
12	Sci-Fi
13	Documentary
14	War
15	Musical
16	Mystery
17	Western
18	Film-Noir

[1] https://grouplens.org/datasets/movielens/

Table 2 presents the different movie categories. In total there are 18 different categories. Therefore, the experimental dataset used for the encoding and testing of the proposed system is a matrix of 3883 movies in rows times 18 movie categories in columns.

4.3.2.2.3 Experimental Results

Three users with different characteristics who searched for various items were used to test the proposed recommendation schema on a movie dataset. As previously mentioned the system utilizes an entropy-based approach and recommends movies based on Hard and Fuzzy K-Modes clustering. The different characteristics used by the system to predict accurate recommended items were: (i) the total number of the final clusters based on the threshold similarity value β, (ii) the ranking of the movie categories according to the user interests and, (iii) the past history of different searches that each of the users conducted. The Root Mean Square Error (RMSE) was used as the evaluation metric to assess the accuracy of the results and is defined in equation (12):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_{obs,i} - x_{model,i})^{2}}{n}} \tag{12}$$

where $x_{\mathrm{obs,i}}$ and $x_{\mathrm{model,i}}$ are the observed and modeled values at the *i-th* sample respectively.

Table 3 shows how one of our users ranked the different movie categories based on his/her preferences. The static information was inserted into the clusters and changed them to include the weight reflecting the interests of the user.

Table 3: UserA Movie Rankings

1. Drama	10. Crime
2. Adventure	11. Mystery
3. Animation	12. Thriller
4. Action	13. Documentary
5. War	14. Romance
6. Children	15. Film-Noir
7. Musical	16. Sci-Fi
8. Comedy	18. Horror
9. Western	19. Fantasy

Table 4 shows the ten first searches that UserA conducted in order for the system to understand his/her behavior and identify the movie categories searched more frequently. If the number searched on a specific category exceeds a specific value, in our case this value is equal to three, then the system selects that category as a “frequently searched”. In the case of UserA the categories searched more are Adventure and Drama. After finalizing the ten first searches the system starts to recommend items to the user based on how he/she ranked the movie categories, what was searched more frequently and the searched keywords. The analysis of the specific searches follows.

Table 4: UserA first 10 searches

ID	Movie Title	Movie Categories
23	Assasins	Thriller
31	Dangerous Minds	Drama
86	White Squall	Adventure , Drama
165	The Doom Generation	Comedy , Drama
2	Jumanji	Adventure, Children, Fantasy
167	First Knight	Action, Adventure, Drama, Romance
237	A Goofy Movie	Animation, Children, Comedy, Romance
462	Heaven and Earth	Action, Drama, War
481	Lassie	Adventure, Children
1133	The Wrong Trousers	Animation, Comedy

Close inspection of the results listed in Table 5 reveals the following findings:

i. The proposed fuzzy algorithm is more accurate on average than the Hard implementation as it adjusts dynamically the knowledge gained by the RS engine.

ii. Both algorithms always suggest movies in ascending order of error magnitude, while there are also cases where movies bear the exact same RMSE value; this is quite natural as they belong to the same cluster with the same degree of membership.

iii. The error depends on the category of the movie searched for each time; therefore, when this type perfectly matches the clustered ones the error is minimized.

Table 5: UserA analytical recommendation results and evaluations per clustering method and search conducted

Searches and Clustering Methods		**Recommendations** 1	2	3	4	5
#1: Dadetown Documentary	**Hard** 0.3333	Two Family House Drama 0.3333	Tigerland Drama 0.3333	Requime for a Dream Drama 0.3333	Remember the Titans Drama 0.3333	Girlfight Drama 0.3333
	Fuzzy 0.3333	Othello Drama 0.3333	Now and Then Drama 0.3333	Angela Drama 0.3333	Dangerous Minds Drama 0.3333	Restoration Drama 0.3333
#2: A Christmas Tale Comedy and Drama	**Hard** 0.0	Bootmen Comedy and Drama 0.0	Beautiful Comedy and Drama 0.0	Duets Comedy and Drama 0.0	A Knight in New York Comedy and Drama 0.0	Mr.Mom Comedy and Drama 0.0
	Fuzzy 0.0	Waiting to Exhale Comedy and Drama 0.0	To Die For Comedy and Drama 0.0	Kicking and Screaming Comedy and Drama 0.0	Big Bully Comedy and Drama 0.0	Nueba Yol Comedy and Drama 0.0
#3: Yellow Submarine Animation and Musical	**Hard** 0.4588	Digimon Adventure, Animation and Children 0.4082	For the Love of Benji Adventure and Children 0.4714	The Legend of Lobo Adventure and Children 0.4714	Tall Tale Adventure and Children 0.4714	Barneys Adventure and Children 0.4714
	Fuzzy 0.4059	Pete's Dragon Adventure, Animation, Children and Musical 0.3333	Gullivers Travels Adventure, Animation and Children 0.4082	Digimon Adventure, Animation and Children 0.4082	Bedknobs and Broomsticks Adventure, Animation and Children 0.4082	The Lord of the Rings Adventure, Animation, Children and Sci-Fi 0. 4714
#4: Young Guns Action, Comedy and Western	**Hard** 0.2576	Butch Cassidy Action, Comedy and Western 0.2576	Action Jackson Action and Comedy 0.2357	Last Action Hero Action and Comedy 0. 2357	Mars Attack Action, Comedy, Sci-Fi and War 0.4082	Tank Girl Action, Comedy, Musical, Sci-Fi 0.4082
	Fuzzy 0.2357	I Love Trouble Action and Comedy 0.2357	Beverly Hills Cop III Action and Comedy 0.2357	The CowBoy Way Action and Comedy 0.2357	Beverly Hills Ninja Action and Comedy 0.2357	Last Action Hero Action and Comedy 0.2357

Table 6 summarizes the mean RMSE values for four additional users tested after conducting five different searches. We omitted the details of the searched categories as these resemble the ones presented for UserA. It is once again clear that the algorithm behaves successfully, with average error values below 0.5 with only one exception (first search of UserB) and with consistently better performance being observed for the fuzzy implementation.

Table 6: Summary of recommendation results and mean evaluations per clustering method for four more users

User	Method	Searches				
		1	***2***	***3***	***4***	***5***
B	Hard	0.5774	0.3480	0.4572	0.4557	0.2357
	Fuzzy	0.5360	0.3368	0.4082	0.4335	0.2552
C	Hard	0.3135	0.3333	0.3437	0.3714	0.2357
	Fuzzy	0.2552	0.2357	0.3437	0.2747	0.2357
D	Hard	0.0	0.4082	0.3333	0.3999	0.4461
	Fuzzy	0.0	0.4082	0.3333	0.3908	0.4082
E	Hard	0.3782	0.3610	0.3714	0.4885	0.2943
	Fuzzy	0.3782	0.3333	0.3333	0.4673	0.2943

4.3.3 Dynamic Web Recommender Systems: A different approach

4.3.3.1 Introduction

The system proposed in this part of our research extends the work previously presented and makes recommendations based on the preferences of the interested user, which are dynamically adjusted in real-time taking into account her/his previous searches using a different number of attributes. Unlike the previous methodology, the new approach is not enhanced by the utilization of static preferences declared by the user when registering into the system, but it now relies on a learning mode for the system's newly registered users. According to this process, the system records their preferences and the way of searching for a number of searches (learning period) and then it starts recommending items. The heart of the recommendation engine is the clustering procedure; thus, the techniques mentioned in the Technical Background section are applied. The proposed system was tested using the MovieLens1M dataset, which was again linked with IMDB.com to retrieve more content information. The experimentation phase involved searches on stars, categories, production companies and any combination between them. The final results indicate that the proposed system meets the design objectives as it delivers items which are closely related to what the user would have liked to receive based on her/his pst behavior.

4.3.3.2 Problem Formulation

Our task is to recommend accurate movies back to the user based not only on his/her preferences and past searching history, but also on the search input captured in real-time. There are three sets in the proposed system:

C: Movie Categories

S: Movie Stars

P: Movie Production Companies

R is defined as a set of recommendations ($r_{i,u}$), where i=1…K is the number of recommended items and $u = 1 \ldots N$ the number of users.

A recommendation is defined as a function that combines information from the three sets mentioned above with that of a number of past searches, that is,

$r_{i,u} = f\,(< C, S, P >, s_{j,u})$, where $j = 1 \ldots S$ the number of past searches of user u.

Our goal is to provide recommendations such that the RMSE value is minimized.

4.3.3.3 Methodology and Dataset

4.3.3.3.1 Proposed System

Figure 2 presents a schematic representation of the proposed system. First, a user registers into the system and starts conducting searches. The first ten searches (this number can be adapted by the user) are utilized in a so-called learning mode of the system, that is, a session to understand what her/his preferences are and how he/she is conducting the searches. After the learning session is over, the system starts recommending items back to the user.

During this process dynamic bit string tables that include the user preferences (one for the movie categories, one for the stars and one for the production companies) are created using the information acquired from the learning session and are updated each time a user conducts a new search. The last three movies (newest searches) are assigned extra (higher) weights and when a user is searching for a specific star, category, production company, or combinations of them, there is also an extra weight for that specific search as shown in Table 7; this is useful for acknowledging the fact that the user was looking for specific parts of the available dataset and use these parts in future recommendations.

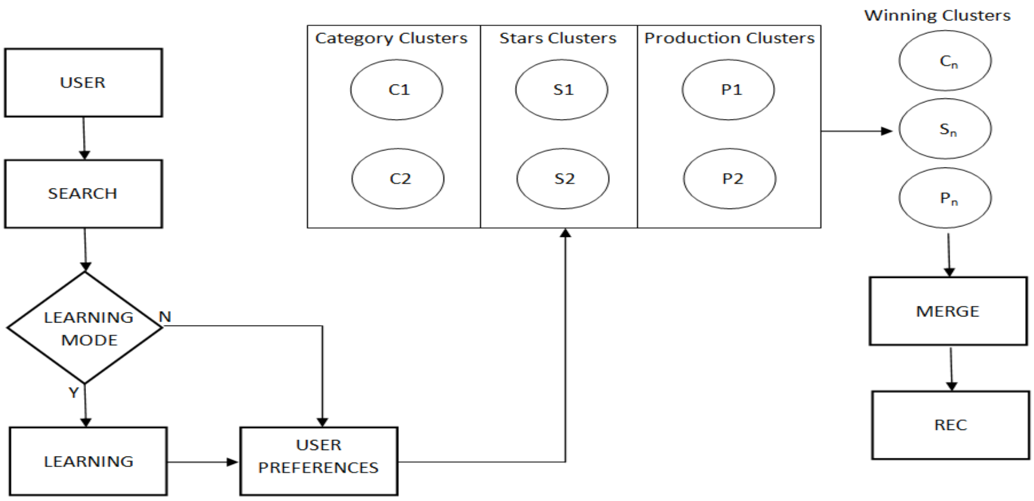

Figure 2: A workflow describing how the proposed RS works

In the end the system computes the weights for all searches in every column and divides them by a specific number that is set from the beginning (this number can be modified) to produce the new dynamic information tables with the new weights. By doing this the system captures users' interest in real-time by taking into consideration his/her latest actions. In the case that a user lost his/her interest on a specific category, star or production company, the system can identify this from his/her latest searches and can quickly adjust its recommendations.

Table 7: Extra weights assigned according to previous searches

8^{th} Movie	multiply bits by 1
9^{th} Movie	multiply bits by 2
10^{th} Movie	multiply bits by 3
Specific Search	multiply specific bit by 5

Afterwards, the system uses the Entropy-based algorithm and the threshold similarity value β so as to find the number of clusters and the clusters' centers. The value of β needs to be tuned first based on the size of the available dataset in order to reach optimal performance; then, this value is treated as constant in order to find the optimal number of clusters and the clusters' centers for each one of the datasets (categories, stars and production companies) that will be used for the clustering techniques.

Continuing, the clustering algorithms of Hard and Fuzzy K-Modes are executed to form the clusters. Three different clustering processes are thus performed, one for the movie categories, one for the stars (actors/actresses) and one for the production companies, each resulting a different number of clusters. The next step involves the update of those clusters to include the dynamic information weights of the particular user.

When a user performs searching the keyword used (input) is compared with each cluster center and the system finds the most similar one (winning cluster) depending on the item searched. In the meantime the keyword searched is saved by the system as part of the dynamic information which is thus updated in real time. In such a case we have three winning clusters, one from the categories, one from the stars and one from the production companies. Therefore, in the end the three clusters are merged and their combined information is used to provide the most accurate recommendations back to the user.

4.3.3.3.2 Dataset

The dataset used by the proposed system in this part of our thesis is the same like the one presented in our previous work; but in this time the final dataset was linked with IMDB.com to retrieve more content information not only regarding a movie's categories, but also about the stars and the production companies of each movie.

The experimental dataset for the movie categories was a matrix of 3883 movies in rows times 18 movie categories in columns as shown in Table 2. Accordingly, the experimental dataset for the movie stars used was a matrix of 3883 movies in rows times 373 movie stars in columns we only provide the first 20 of them in Table 8.

Table 8: Movie starts/actors

Column	Movie Star
1	Tom Hanks
2	Robin Williams
3	Walter Matthau
4	Whitney Houston
5	Steve Martin
6	Al Pacino

7	Harrison Ford
8	Jonathan Taylor
9	J-Claude Van Damme
10	Pierce Brosnan
11	Michael Douglas
12	Leslie Nielsen
13	Kevin Bacon
14	Anthony Hopkins
15	Geena Davis
16	Robert De Niro
17	Emma Thompson
18	Tim Roth
19	Jim Carrey
20	Wesley Snipes

The corresponding list for the movie production companies is shown in Table 9 (again 20 listings) and the associated dataset consists of 3883 movies in rows multiplied by 152 movie production companies in columns.

Table 9: Movie production Companies

Column	Production Company
1	Pixar
2	Walt Disney
3	TriStar
4	Warner Bros
5	MGM
6	Lancaster
7	Universal
8	Columbia
9	Paramount
10	Canal+
11	New Line Cinema
12	Miramax
13	Castle Rock Ent.
14	Lumiere
15	Morgan Creek
16	PolyGram
17	Mirage
18	Atlas Entertainment
19	Hollywood Pictures
20	Caravan

Summarizing, the proposed system uses three different datasets to structure the information for the dynamic information tables (user preferences) and produce the final clusters. The datasets are linked together using the Movie ID.

4.3.3.4 Experimental Process

4.3.3.4.1 Design of Experiments

Five users with different characteristics and behaviors that conducted searches on the dataset were used as case studies to assess the proposed recommendation schema. As previously mentioned, the system recommends movies based on the Hard and Fuzzy K-Modes clustering.

The different characteristics used by the system to accurately predict and recommend movies were: (i) the total number of the final clusters for each dataset based on the threshold similarity value β, (ii) the searches conducted by each user in the learning mode (past history) and, (iii) the new searches combining movie categories, stars and production companies.

Again in this part of our research, the Root Mean Square Error (RMSE) was used as the evaluation metric to assess the accuracy of the results. Three different RMSE values that compare the search input with the recommendations are actually calculated, one for the movie categories, one for the stars and one for the production companies; then, their average value is taken as the total error.

4.3.3.4.2 Experimental Results

Table 10 shows the analytical searches of one of our test users named “UserA”. The user conducted a number of searches at the learning mode in order for the system to understand its behavior by identifying which movie categories, stars and production companies the user searched more. This information was inserted into the dynamic information tables and was continuously updated using the information of the new searches.

The system ranks the category, star, or production company from “frequently searched” to “rarely searched” based on the weights formed in the dynamic information tables.

Table 10: Learning session for UserA

Movie ID	Movie Title	Movie Categories	Movie Stars	Movie Production Companies
1599	I Know What You Did Last Summer (1997)	Horror, Mystery and Thriller	Jennifer Love Hewitt, Sarah Michelle Gellar and Anne Heche	Columbia Pictures Corporation, Mandalay Entertainment and Summer Knowledge LLC
1664	Midnight in the Garden of Good and Evil (1997)	Comedy, Crime, Drama and Mystery	John Cusack, Kevin Spacey and Jack Thompson	Malpaso Productions, Silver Pictures and Warner Bros. Pictures
32	Twelve Monkeys (1995)	Drama and Sci-Fi	Bruce Willis, Madeleine Stowe and Brad Pitt	Universal Pictures, Atlas Entertainment and Classico
50	The Usual Suspects (1995)	Crime and Thriller	Kevin Spacey, Gabriel Byrne and Chazz Palminteri	PolyGram Filmed Entertainment, Spelling Films International, Blue Parrot, Bad Hat Harry Productions and Rosco Film GmbH
86	White Squall (1996)	Adventure and Drama	Jeff Bridges, Caroline Goodall and John Savage	Hollywood Pictures, Largo Entertainment and Scott Free Productions
119	Race the Sun (1996)	Drama	Halle Berry, James Belushi and Casey Affleck	TriStar Pictures, American Broadcasting Company (ABC), Columbia TriStar
272	Man of the House (1995)	Comedy	Chevy Chase, Farrah Fawcett and Jonathan Taylor Thomas	All Girl Productions, Forever Girls Productions, Marty Katz Productions, Orr & Cruickshank and Walt Disney Pictures
469	I'll Do Anything (1994)	Comedy and Drama	Nick Nolte, Albert Brooks and Whittni Wright	Columbia Pictures Corporation and Gracie Films
511	The Ref (1994)	Comedy	Denis Leary, Judy Davisand Kevin Spacey	Don Simpson/Jerry Bruckheimer Films and Touchstone Pictures
439	Demolition Man (1993)	Action and Sci-Fi	Sylvester Stallone, Wesley Snipes and Sandra Bullock	Warner Bros. Pictures and Silver Pictures

Table 11 provides the searches conducted by User "A" after the conclusion of the learning session, when he/she starts receiving recommendations. Also, Table 12 shows the detailed information of each of the movies searched.

Table 11: How and what UserA searched for

Search Input	Chosen Movie ID	Chosen Movie Title
Tom Hanks and Animation	1	Toy Story (1995)
Crime and Drama	22	Copycat (1995)
Mortal Kombat (1995)	44	Mortal Kombat (1995)
Adventure, Children, Melenny	56	Kids of the Round Table (1995)
Al Pacino, Castle Rock and Columbia	99	City Hall (1996)
Jim Carrey and Comedy	152	Batman Forever (1995)
Horror and United Artists	176	Lord of Illusions (1995)
Safe (1995)	189	Safe (1995)
Romance	234	French Kiss (1995)
Sarah Jessica Parker, Antonio Banderas	276	Miami Rhapsody (1995)

Table 12: Detailed information of the movies UserA searched

Movie Title	Movie Categories	Movie Stars	Movie Production Companies
Toy Story (1995)	Animation, Children and Comedy	Tom Hanks, Tim Allen and Don Rickles	Pixar and Walt Disney
Copycat (1995)	Crime, Drama and Thriller	Sigourney Weaver, Holly Hunter and Dermot Mulroney	Regency Enterprises and New Regency Picture
Mortal Kombat (1995)	Action and Adventure	Christopher Lambert, Robin Shou and Linden Ashby	New Line Cinema and Threshold Entertainment
Kids of the Round Table (1995)	Adventure, Children and Fantasy	Johnny Morina, Maggie Castle and Christopher Olscamp	Melenny Productions and Téléfilm Canada
City Hall (1996)	Drama and Thriller	Al Pacino, John Cusack and Bridget Fonda	Castle Rock Entertainment and Columbia Pictures Corporation
Batman Forever (1995)	Action, Adventure, Comedy and Crime	Val Kilmer Tommy Lee Jones and Jim Carrey	Warner Bros. Pictures and PolyGram Filmed Entertainment
Lord of Illusions (1995)	Horror	Scott Bakula, Kevin J. O'Connor and Joseph Latimore	United Artists and Seraphim Films
Safe (1995)	Thriller	Julianne Moore, Xander Berkeley and Dean Norris	American Playhouse Theatrical Films, Killer Films, Chemical Films, Good Machine, Kardana Productions, Channel Four Films, Arnold Semler, American Playhouse and Kardana Films
French Kiss (1995)	Comedy and Romance	Meg Ryan, Kevin Kline and Timothy Hutton	Polygram Filmed Entertainment, Prufrock Pictures, Twentieth Century

			Fox Film Corporation and Working Title Films
Miami Rhapsody (1995)	Comedy	Sarah Jessica Parker, Mia Farrow and Antonio Banderas	Cantaloupe Production and Hollywood Pictures

Close inspection of the results listed in Table 13 reveals the following findings:

iv. The two proposed algorithms present quite similar performance, with the fuzzy algorithm being slightly more accurate on average than the Hard implementation as it adjusts dynamically the knowledge gained by the RS engine.

v. Both algorithms always suggest movies in ascending order of error magnitude, while there are also cases where movies bear the exact same RMSE value; this is quite natural as they belong to the same cluster with the same degree of membership.

vi. The error depends on the category of the movie searched for each time; therefore, when this type perfectly matches the clustered ones the error is minimized.

Table 13: Recommendation results and evaluations per clustering method and search conducted by User "A"

Searches and Clustering Methods		**Recommendations**				
		1	***2***	***3***	***4***	***5***
#1:Toy Story (1995)	**Hard** 0.0829	Toy Story 2 (1999) 0.0244	A League of Their Own (1992) 0.2128	Chicken Run (2000) 0.0813	A Bug's Life (1998) 0.0345	Aladdin and the King of Thieves (1996) 0.0615
	Fuzzy 0.1456	Toy Story 2 (1999) 0.0244	Forrest Gump (1994) 0.2283	A League of Their Own (1992) 0.2128	Splash (1984) 0.2283	A Bug's Life (1998) 0.0345
#2:Copycat (1995)	**Hard** 0.0919	Dog Day Afternoon (1975) 0.1726	Kiss the Girls (1997) 0.0727	Once Upon a Time in America (1984) 0.0656	Guilty as Sin (1993) 0.0681	Kiss of Death (1995) 0.0805
	Fuzzy 0.0684	The Thomas Crown	Kiss the Girls (1997)	Desperate Measures (1998)	Once Upon a Time in America (1984)	Kiss of Death (1995)

		Affair (1968) 0.0615	0.0727	0.0615	0.0656	0.0805
#3: Mortal Kombat (1995)	**Hard** 0.0663	Mortal Kombat - (1997) 0.0244	Knockout (1999) 0.0681	Indiana Jones and the Temple of Doom (1984) 0.0813	Kull the Conqueror (1997) 0.0767	Raiders of the Lost Ark (1981) 0.0813
	Fuzzy 0.0683	Mortal Kombat - (1997) 0.0244	Let's Get Harry (1986) 0.0767	Allan Quartermain and the Lost City of Gold (1987) 0.0768	The Poseidon Adventure (1972) 0.0813	Indiana Jones and the Last Crusade (1989) 0.0813
#4: Kids of the Round Table (1995)	**Hard** 0.0789	Santa Claus - The Movie (1985) 0.0681	The Never Ending Story (1984) 0.0767	The Goonies (1985) 0.0839	Escape to Witch Mountain (1975) 0.0767	The Never Ending Story III (1994) 0.0891
	Fuzzy 0.0826	Santa Claus - The Movie (1985) 0.0681	Seventh Heaven (1997) 0.0767	Labyrinth (1986) 0.0767	The Never Ending Story III (1994) 0.0891	The Indian in the Cupboard (1995) 0.1027
#5: Cityhall (1995)	**Hard** 0.1481	Looking for Richard (1996) 0.1924	Beyond Rangoon (1995) 0.1456	Before Sunrise (1995) 0.1410	In the Line of Fire (1993) 0.1533	The Run of the Country (1995) 0.1084
	Fuzzy 0.1945	Amos & Andrew (1993) 0.2017	Honeymoon in Vegas (1992) 0.2376	The Tingler (1959) 0.1659	The Godfather (1972) 0.2073	Glengarry Glen Ross (1992) 0.1599
#6: Batman Forever (1995)	**Hard** 0.2126	Dumb & Dumber (1994) 0.2145	Ace Ventura - When Nature Calls (1995) 0.2128	Man on the Moon (1999) 0.2356	Liar Liar (1997) 0.2001	The Cable Guy (1996) 0.2001
	Fuzzy 0.2101	Liar Liar (1997) 0.2001	Man on the Moon (1999) 0.2356	Me, Myself and Irene (2000) 0.2001	Dumb & Dumber (1994) 0.2145	The Cable Guy (1996) 0.2001
#7: Lord of Illusions	**Hard** 0.0570	Red Dawn (1984) 0.1360	Licence to Kill (1989) 0.1111	The Rage - Carrie 2 (1999) 0	The Three Ages (1923) 0	Castle Freak (1995) 0.03823

	Fuzzy 0.0671	Licence to Kill (1989) 0.1111	The Rage - Carrie 2 (1999) 0	Rocky V (1990) 0.1533	The Fear (1995) 0.0270	Cronos (1992) 0.0442
#8: Safe (1995)	**Hard** 0.0749	The Boys from Brazil (1978) 0.0640	In Dreams (1999) 0.0903	Office Killer (1997) 0.0640	Trial by Jury (1994) 0.0784	The Tie That Binds (1995) 0.0777
	Fuzzy 0.0768	Number Seventeen (1932) 0.0712	Stranger in the House (1997) 0.0713	The 39 Steps (1935) 0.0713	Fear (1996) 0.0713	Malice (1993) 0.0990
#9: French Kiss (1995)	**Hard** 0.0603	The Closer You Get (2000) 0.0442	Fever Pitch (1997) 0.0554	Jeanne and the Perfect Guy (1998) 0.0713	The Butcher's Wife (1991) 0.0681	Michael (1996) 0.0626
	Fuzzy 0.0487	That Old Feeling (1997) 0.0442	When Harry Met Sally (1989) 0.0554	Strictly Ballroom (1992) 0.0442	Ninotchka (1939) 0.0442	Rendezvous in Paris (1995) 0.0554
#10: Miami Rhapsody (1995)	**Hard** 0.1870	Assassins (1995) 0.1950	If Lucy Fell (1996) 0.1552	Evita (1996) 0.1875	Mars Attacks (1996) 0.1987	Extreme Measures (1996) 0.1987
	Fuzzy 0.1479	Women on the Verge of a Nervous Breakdown (1988) 0.1228	Play it to the Bone (1999) 0.1498	Two Much (1996) 0.1340	Mars Attacks (1996) 0.1987	Hocus Pocus (1993) 0.1340

Table 14 summarizes the RMSE error values for four additional users tested on ten different searches. It is again evident that the fuzzy approach is slightly superior to the hard implementation.

Table 14: Summary of recommendation results and mean evaluations per clustering method for four more users

User	Method	1	2	3	4	5	Average
A	Hard	0.0829	0.0919	0.0663	0.0789	0.1481	***0.0936***
	Fuzzy	0.1456	0.0684	0.0683	0.0826	0.1945	***0.1118***
B	Hard	0.0820	0.0648	0.1639	0.0932	0.1970	***0.1201***
	Fuzzy	0.0755	0.0308	0.1973	0.0932	0.1290	***0.1051***
C	Hard	0.1870	0.2128	0.1341	0.0668	0.1280	***0.1457***
	Fuzzy	0.1831	0.1404	0.1923	0.0779	0.1284	***0.1444***
D	Hard	0.1081	0.1762	0.1970	0.1976	0.1861	***0.1730***
	Fuzzy	0.0865	0.1150	0.1994	0.2494	0.1699	***0.1640***
E	Hard	0.0681	0.1453	0.1085	0.1133	0.1706	***0.1211***
	Fuzzy	0.0749	0.0866	0.0818	0.1640	0.1608	***0.1136***
		6	**7**	**8**	**9**	**10**	**Average**
A	Hard	0.2126	0.0570	0.0749	0.0603	0.1870	***0.1183***
	Fuzzy	0.2101	0.0671	0.0768	0.0487	0.1479	***0.1101***
B	Hard	0.0549	0.0682	0.1418	0.0581	0.1171	***0.0880***
	Fuzzy	0.0560	0.1225	0.1164	0.0654	0.0759	***0.0872***
C	Hard	0.0371	0.1952	0.0998	0.2109	0.0896	***0.1265***
	Fuzzy	0.0298	0.2475	0.1272	0.2295	0.1121	***0.1492***
D	Hard	0.2157	0.1389	0.1805	0.1792	0.0759	***0.1580***
	Fuzzy	0.2327	0.1440	0.1753	0.1292	0.0788	***0.1520***
E	Hard	0.1673	0.1760	0.0521	0.0917	0.1287	***0.1231***
	Fuzzy	0.1541	0.1770	0.0447	0.1141	0.1100	***0.1199***

Figures 3 and 4 show the RMSE values for each of the users and for the two clustering methods, Hard K-Modes and Fuzzy K-modes. It is once again clear that both algorithms behave successfully, with average error values below 0.25 for all cases reported.

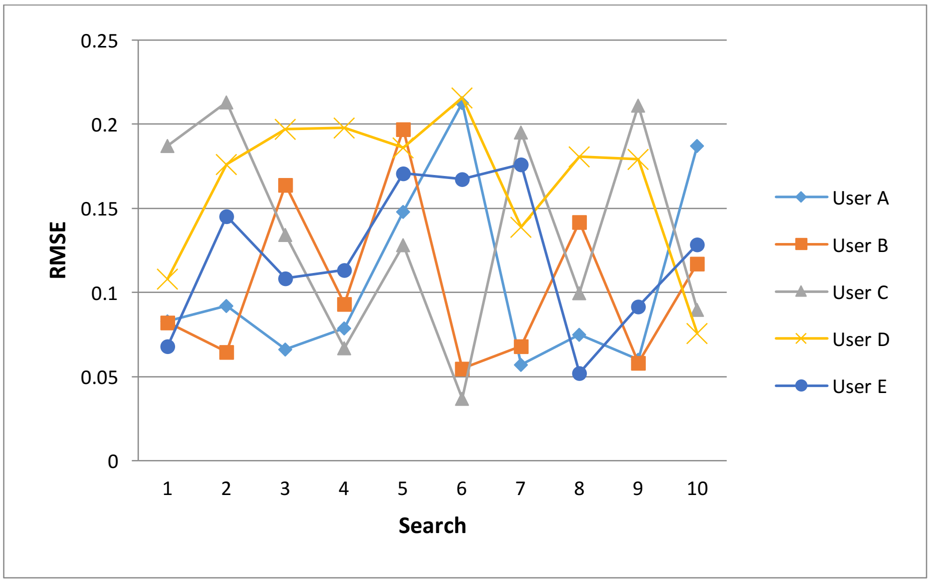

Figure 3: Hard K-Modes RMSE values for all users and their searches

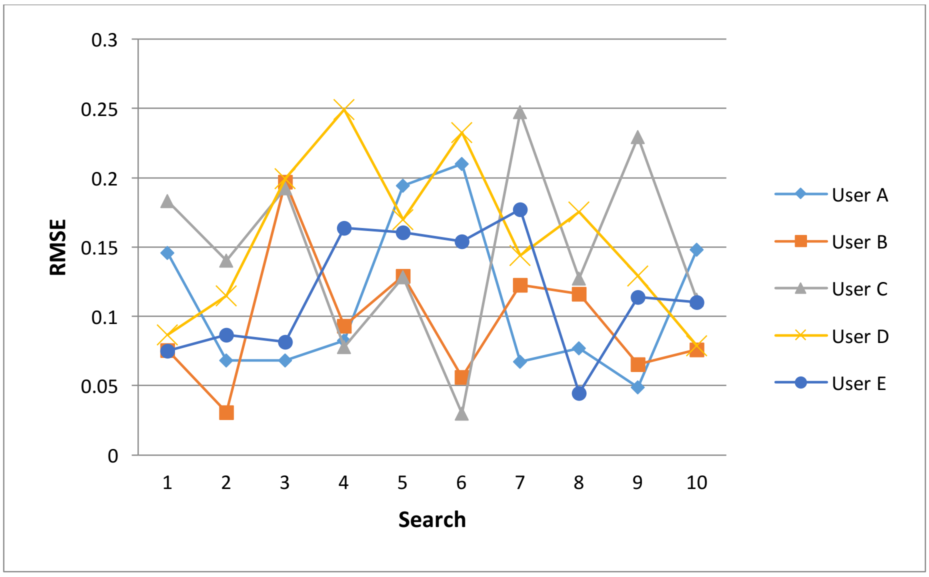

Figure 4: Fuzzy K-Modes RMSE values for all users and their searchers

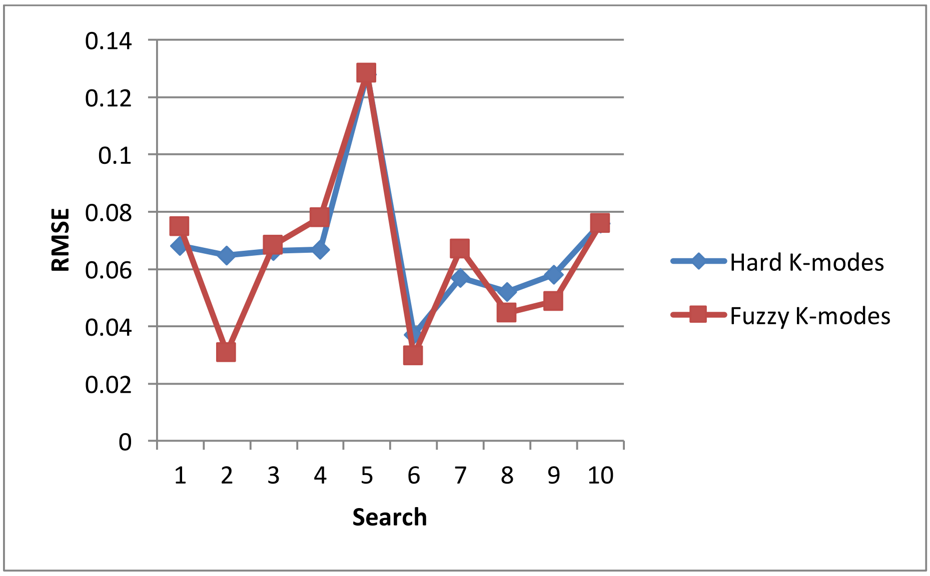

Figure 5: Hard K-Modes vs Fuzzy K-Modes best accuracies per search

Figure 5 presents the best RMSE values between users per search for each one of the proposed clustering methods. Taking into consideration the average RMSE for every 5 searches as shown in Table 14 it's clearly shown that the Fuzzy K-modes outperforms the Hard K-Modes by predicting results with higher accuracy as the system learns the preferences of the user.

4.3.3.4.3 Comparison with the k-NN algorithm

Two implementations of the well-known k-nearest neighbor algorithm were implemented and executed with the available dataset for comparison purposes. The first is the simple k-NN case according to which the following steps are followed:

- Step 1: Find the 5 closest neighbors with regards to the search input using the Hamming distance and return them back to the user as the result of his/her search
- Step 2: Find the 5 closest neighbors of the neighbours of the previous point (this results in 25 movies)
- Step 3: Multiply the relevance of movies with weights of the learning session (ten first searches)
- Step 4: Sort in descending order and select the top 5 scores as recommendations

The second variation is called k-NND and is somewhat more "sophisticated" than the simple algorithm. It essentially combines the classic approach with the dynamic information used in the proposed RS, which is multiplied by additional weights formed as searches progress right after the execution of step 3 above.

Both implementations of the k-NN algorithm were executed for the “User A” case study, the results are given in Table 15 and graphically depicted in Figure 6. It is quite clear that the proposed clustering approaches performed better compared to both k-NN implementations. Moreover, we can safely argue that incorporating the dynamic information increases the accuracy of the recommendations as this is evident in the RMSE behavior of the k-NND (KNN-D) compared to the simple k-NN implementation (KNN). Therefore, the basic principle of the proposed approach, that is, to rely on information that may constantly change, is proved the cornerstone for its successful performance.

Table 15: Comparison of the proposed clustering techniques with the k-NN algorithms for “UserA”

Method	**Searches**									
	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
KNN	0.2457	0.1464	0.2100	0.2609	0.2160	0.2377	0.1970	0.2085	0.1865	0.2095
KNN-D	0.1554	0.0745	0.1802	0.0831	0.1503	0.2202	0.1749	0.1847	0.1714	0.1961
Hard	0.0829	0.0919	0.0663	0.0789	0.1481	0.2126	0.0570	0.0749	0.0603	0.1870
Fuzzy	0.1456	0.0684	0.0683	0.0826	0.1945	0.2101	0.0671	0.0768	0.0487	0.1479

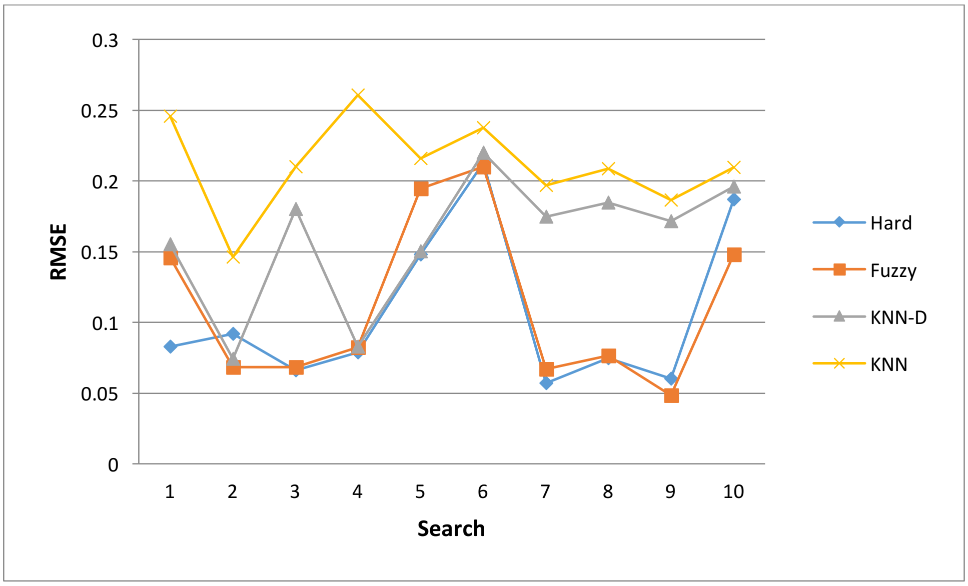

Figure 6: Hard and Fuzzy Clustering vs two k-NN implementations

4.3.4 Overview

This section presents a series of RS that use multiple-criteria aiming to produce accurate recommendations to users and to tackle at the same time the data sparsity, cold-start problem as well any scalability issues. The first approach is enhanced by the utilization of static preferences declared by the user when entering a website in order for the RS to build its profile and the second approach relies on a learning session where the system records the users' preferences while conducting searches on the platform and then starts the recommendation procedure. The proposed methodologies differ from the usual *users x items x ratings* approach as now the system builds a user profile during a learning session instead of requiring any ratings on items to produce recommendations. The methodologies described in this section utilize various clustering methods and the final results indicate that the proposed models perform better than other baselines methods. A final outcome that can be drawn out is that Fuzzy K-Modes approach outperforms the Hard K-Modes method in terms of accuracy.

The work outlined in this section lead to the publication of one conference paper (Christodoulou et al., 2013) and one journal paper (Christodoulou et al., 2014).

4.4 A Real-time targeted Recommender System

The second part of the Multi-criteria RS presents a framework for deploying a RS in a store/shop environment that aims to suggest real-time personalized offers to customers. Stores' customers find it difficult to choose from a large variety of products or be informed about the latest offers that exist in a shop based on the items that they need or wish to purchase. In this framework, as customers navigate in a store, various devices like iBeacons push personalized notifications to their smart-devices informing them about offers that are likely to be of interest.

The general structure described in this part uses datasets that contain multiple characteristics. The proposed system utilizes the methodologies presented in the Technical background section. The Entropy-based algoirthmaims to determine the number of clusters and the clusters' centers from a dataset of registered users with different preferences (static information). The Hard K-modes clustering algorithm is used to group users with similar characteristics based on their preferences. As in the previous apaorach, the aforementioned methodologies overcome the cold-start problem due to the fact that now every new user is assigned into a cluster, and at the same time minimize any scalability issues.

A rule-based system is applied to create personalized sub-datasets of products for each cluster reducing the search space in the overall set of products dealing with scalability issues and also with the data Sparsity problem as the system computes recommendations by taking into consideration only the set of products of a certain cluster.

The probabilistic model described in this framework utilizes the users' transaction history to learn their frequent shopping habits; these practices are then used as input to a Bayesian Inference approach to determine whether a product is suitable for purchase or not (Christodoulou et al., 2015). The main goal of the proposed recommendation engine is to suggest targeted products from a list of products that are on offer to users in real-time using iBeacons or any other suitable technologies. The product recommendations are sent to users while they navigate into a store reducing also the time needed on a shelf to find the best offer that suits them. Furthermore, the proposed approach introduces additional features that enhance a customer's shopping experience. Finally, the proposed framework calculates the increase in the store's revenue when customers buy products that appear in their recommendation list.

The proposed recommendation engine was tested using a real-world and a synthetic dataset and the final results indicate an increase on the system's accuracy compared with classic collaborative filtering methods.

4.4.1 Related Work

The work in (Lawrence et al., 2001) contributes a personalized RS that suggests new products to supermarket buyers based on association rules. Association rules are applied to define the relationships among the products. A clustering technique is then utilized to cluster shoppers with similar spending histories. An increase in the supermarket's revenue is observed when shoppers choose to purchase products from the recommendation list.

Modeling temporal dynamics is another parameter that influences present RS due to the fact that users tend to change their preferences over time. Users' behavior can be defined by short-term and long-term preferences (Xiang et al, 2010). According to (Yang et al.2015), product attention and popularity are regularly changing making customers reconsider their tastes, interests and feelings. Time changing behavior on the data is therefore an important factor to be taken into consideration when designing a RS.

The work in (Suksom et al., 2010) proposes a personalised food RS based on a rule-based approach. The system aims to offer personalized recommendations on different kinds of meals to users based on their nutrition requirements or other health care characteristics. The work in (Nikoletic, 2013) implements a RS in physical stores targeting to make shopping more interactive to buyers. The proposed methodology utilizes user-based and item-based CF methods, and a Restricted Boltzmann machine algorithm. The system computes recommendations based on the customers' purchase patterns and uses iBeacons to locate a mobile device in a real-world implementation to push recommendations.

The authors in (Lacic et al., 2015) present a RS based on CF that makes use of a user's location captured by indoor position systems. The RS relies on user-based CF to suggest items with no data available. The authors argue that the proposed method outperforms Matrix Factorization approaches when dealing with cold-start users.

4.4.2 The Proposed Real-time RS Framework

The proposed methodology shown in Figure 7 combines the Entropy-based algorithm and Hard K-modes clustering method. Furthermore, it utilizes a Bayesian Inference approach in order to compute recommendations and notify customers about the best/latest offers based on their shopping preferences. The proposed methodology plans to improve the customer's overall shopping experience by suggesting personalized targeted items with accuracy and efficiency.

Modern stores use loyalty schemes (e.g., loyalty cards) to reward repeated customers on purchases. In this part of our research, we suggest an interactive rewarding scheme guided by a dynamic, real-time targeted recommendation engine. More specifically, we propose the replacement of loyalty cards with an interactive smart-device application, which acts both as a bonus card and as a recommendation engine.

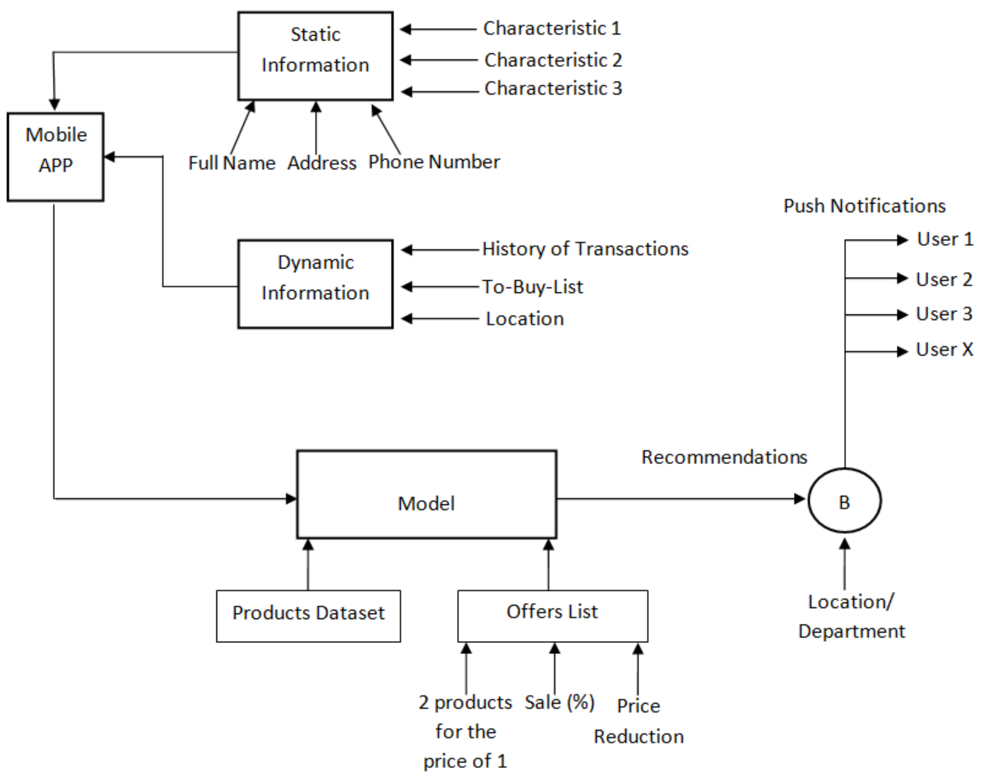

Figure 7: Real-time targeted RS system

A simple mobile application allows users to register or link their existing loyalty account to the system. Figure 8 guides the registration process in which each user is asked to develop its personal profile by providing information regarding the user's Full Name, Telephone Number, Address, Phone Number and other characteristics. As many customers already have a loyalty card, this application allows them to link their existing card with it. In this case, when an existing loyalty card is linked with the mobile application, the customer is asked to provide additional information to build a full profile.

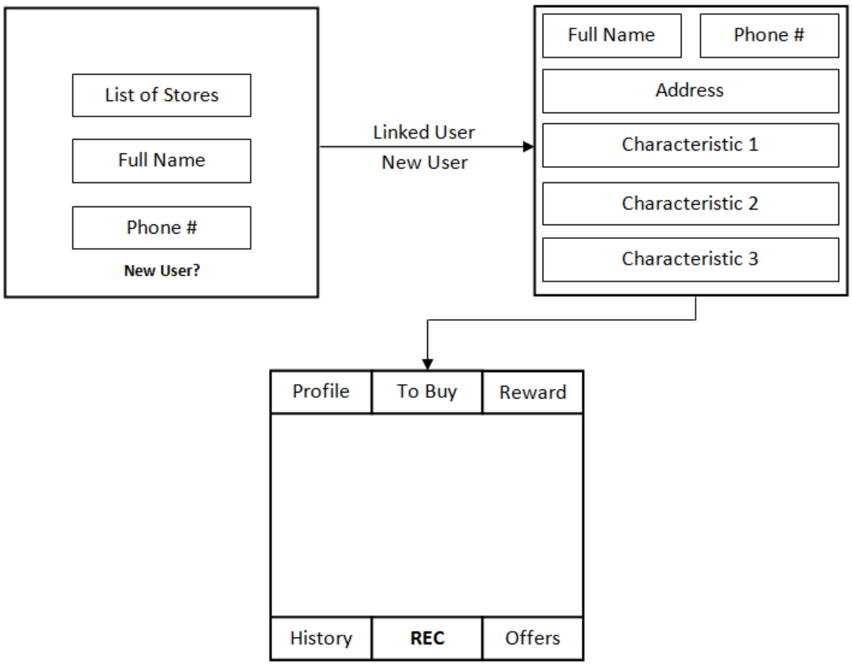

Figure 8: System's mobile application

The following we refer to the information provided by a user as the user's *static information*. Note here that static information is updated dynamically when users manually make changes to their profiles using their mobile application. In addition to user's *static information,* there is also additional information for each user regarding to the user's recent transactions, historical transactions, location of most visited store, products to buy etc. This information changes dynamically in real-time so we refer to it as the user's *dynamic information.* Both static and dynamic information of a profile are used to guide a personalized recommendation process for each user.

The personalized recommendations that are produced by the system are propagated in real-time to each user through iBeacons as a user navigates in the store. In essence, an iBeacon is a Bluetooth low-energy wireless technology developed by Apple that allows mobile applications (iOS or Android) to hear the signals from iBeacons that are in proximity. Broadly speaking, they consist of two processes: (i) the device that broadcasts the data (i.e., iBeacon), and (ii) an application installed on a smart device, which acts as the recipient (Mileta, 2015). In this part the iBeacon technology is used to push personalized recommendations to customers based on information (i.e., static and dynamic) from their profiles.

The user's static information has a twofold purpose it is used: (i) to identify the number of clusters *k* that exist in the dataset of registered users with the system, and (ii) to group each user in a cluster based on his/her preferences. In cases where a user's profile is updated/altered, then the cluster where the user belongs to is updated as well.

In order to maximize the use of the user's static information and to overcome with the challenges that exist in RS, an Entropy-based approach is used to find the different number of clusters and the cluster centers based on users' static information. Moreover, a Hard K-Modes clustering method is utilized to group each user in a specific cluster. This procedure helps the system to deal with the cold-start problem as every new user is assigned to a cluster. Finally, a dedicated rule-based system applies a set of rules to each cluster to create various sub-datasets of products suitable for each group of users aiming to overcome scalability issues as now the system uses only the specific dataset of products for a certain cluster to compute the recommendations and not the overall dataset.

4.4.3 Supermarket Environment

The proposed methodology was applied in a supermarket environment. The static information of the users includes additional information as regards a user's Nationality and Religion, any Nutrition/Diet characteristics, any Allergies on specific products/ingredients, if the user is Vegetarian/Fasting and the current Health Status. As already mentioned, when a user wants to create a new account or link its existing one with the mobile application he/she is required to provide the information that is missing. This information is used by the system to compute the clusters and the clusters' centers and group each user into a specific cluster.

In a supermarket environment a rule-based personalization technique is used for the preprocessing and filtering of the overall dataset of products. This procedure is utilized after classifying users who share the same or similar characteristics in a cluster. The set of rules shown in Figure 9 is used to create sub-datasets of the overall dataset of products that are more suitable for each cluster. To reduce the time needed for computing the clusters and to reduce the computation power needed to derive the recommendations, this procedure is performed off-line.

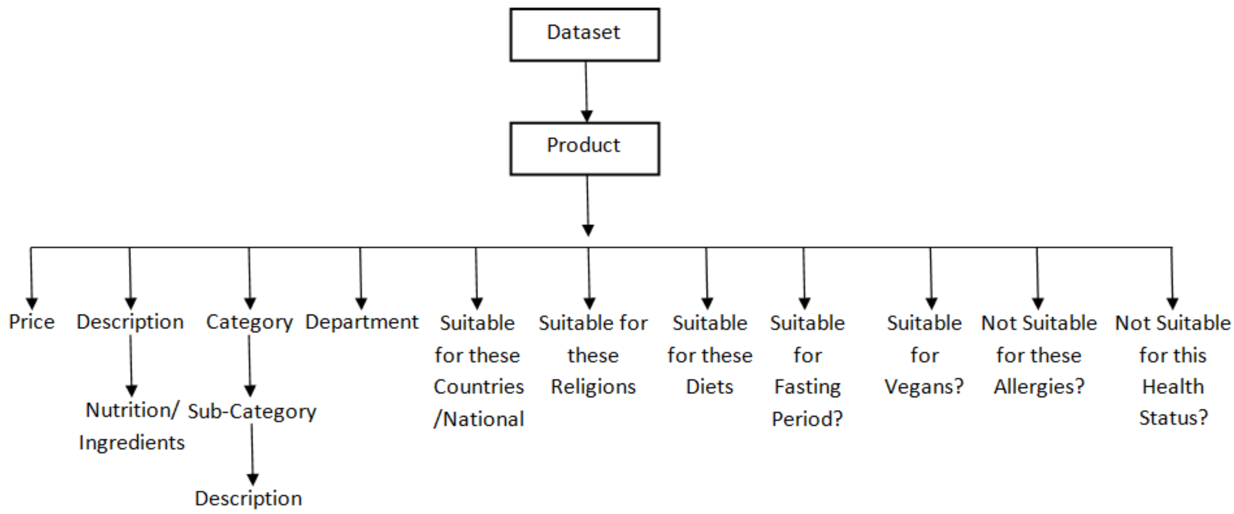

Figure 9: Rule-based system for supermarkets

More specifically, the rules are associated with certain types of information:

1. Country (List of Countries) – The system takes into account the different demographic characteristics and food habits of each country.

2. Nationality (List of Nationalities) – Each nationality has its own characteristics, habits and preferences; the system takes into account such particularities.

3. Religion (List of Religions) – All religions have their own unique characteristics that are considered by the proposed system.

4. Fasting (Enable/Disable) – Some religions have a fasting period, for example during Christmas and Easter for the Christians. If a user enables this option the system determines any fasting habits, as well as, the starting and ending date of each period.

5. Vegetarian (Enable/Disable) – When a user enables this option the system is suggesting products suitable for vegetarians only.

6. Diet (Enable/Disable) – If a user enables this option, the system immediately understands that a user is likely to be on diet or wishes to start a new diet. The system also requires users to specify the type of diet: Low-calories, Low-carbohydrate, High-protein, Low-fat, etc. This information is taken into account during the recommendation stage.

7. Allergies (Enable/Disable) – If this option is enabled it means that a user is suffering from allergies on specific products. The user must choose the specific type of allergy is suffering from using a list that is presented. Each allergy type is linked with products that must be avoided by the user.

    a. Food Allergies (Milk, Egg, Wheat, Nut, Fish, Shellfish, Sulfite, Soy, Casein, Vegetable etc.)

    b. Latex Allergy

    c. Drug Allergy

    d. Skin Allergy

    e. Allergic rhinitis

    f. Other

8. Health status (Enable/Disable) – If this option is enabled the user is asked to provide more information for a list of health indicators.

    a. Cholesterol

    b. High Blood Pressure

    c. Sugar

    d. Depression

    e. Other

The system considers rules that are depended on each other and therefore applies them based on a certain order or a priority. For example, if a user belongs to a religion group that forbids the consumption of beef then this is taken into account.

After clustering the users and finding the appropriate sub-dataset of products for each cluster of users, the system assesses the shopping preferences of each user to determine whether a product is suitable for purchase or not using a Bayesian Inference approach described in the following section. The recommendation engine is configured to make personalized targeted recommendations based on the unique shopping preferences of each user presented.

4.4.4 Probabilistic Model:

Following the clustering of users and the tracing of the appropriate sub-dataset of products for each cluster, the system considers the shopping preferences of each user to determine whether a product on offer is suitable for purchase or not using a Bayesian Inference approach. The Bayesian rule presented in (De Vos., 2008) is described in equation (13):

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)} \tag{13}$$

is observed from the definition of conditional probability as shown in equation (14).

$$P(A \cap B) = P(A/B)P(B) = \mathrm{P(B/A)P(A)} \tag{14}$$

P(A) is the prior: what is known about A before B is observed.

P(B|A) is the likelihood. Note that it only refers to the observed fact B, for all values of A.

P(A|B) is the posterior: what is known about A after observing B.

P(B) can be computed as described in equation (15):

$$P(B) = \int P(A)P(B/A)dA \tag{15}$$

which is a function of *P(A)* and *P(B|A)* to calculate the posterior *P(A/B)*, a new probability statement about A given B.

Bayesian inference presents how to learn from data about an uncertain state of the world. It is determined on rules based on data that might arise for different states of A (De Vos., 2008) and is used to compute the probability for a hypothesis as more evidence or information becomes available.

In this part of our research, a training procedure takes place to estimate the likelihoods by taking into consideration characteristics from the user's transaction history (e.g., frequently bought products of a certain category and average price spent on products from a certain category).

Given a pair of products from the same category/sub-category, a Bayesian Inference approach (Ning et al., 2015) is utilized to derive a probability to determine whether a product is suitable for purchase or not. To reason over such a hypothesis, the model takes into consideration a set of evidence from the customer's transaction history.

To formalize the probabilistic model, we denote with H a Boolean hypothesis on whether a product is suitable for purchase or not. Let $E = \{e_1, e_2 \ldots e_n\}$ denote a sequence of independent evidence inferred from the transaction history. To reason over our hypothesis, we model it as a conditional probability $P(H|E)$ and apply equation (16).

$$P(H/E) = \frac{P(E/H)P(H)}{P(E)} \tag{16}$$

We assume that the prior probability *P(H)* is the degree of belief in judging the hypothesis in the absence of any previous evidence; therefore, we assume a uniform distribution (De Vos., 2008). *P(E)* is the probability of the evidence that is used as a normalization factor and is derived using the law of total probability (De Vos., 2008).

***Training Data**:* To apply equation (16) we estimate the likelihood of observing the evidence (denoted by $P(E|H)$), when the hypothesis is *true* or *false*. To obtain suitable data to train our model the system performs as follows:

Algorithm 1 - Rank frequency bought products

Requires: *User u, Transaction History th*

for all product categories do

rankFrequentProducts(User u, TransactionHistory th) {

1. *For each product category, pc*
   a. *Rank each product using,*

$$frequency(product) = \frac{quantity}{quantityofcategory} \tag{17}$$

end for

Return Ranked List

The steps presented in Algorithm 1 utilizing also equation (17) return a personalized Ranked List (denoted by RK) of Products from each category. Our aim is to recommend products of the same category from the overall dataset of products that are on offer (i.e., “Offer-List”), according to the top-k ranked products from *RK*.

Iteration 1: Evidence is Product Description

At the first iteration the model considers, as a piece of evidence, the string similarity calculated by using a similarity measure (e.g., cosine-similarity (Ning et al., 2015)) between a pair of products based on the products descriptions. This is shown in equation (18).

$$P(H/E_1 = similarityscore) = \frac{P(E_1 = similarityscore/H)P(H)}{P(E_1 = similarityscore)} \tag{18}$$

For the computation of the likelihoods a bootstrapping procedure similar to (De Vos, 2008) is followed for deriving probability distributions from similarity scores.

Iteration 2: Evidence is the average Price of given category

Once a posterior probability is computed for some evidence $e_1 \in E$ a new piece of evidence $e_2 \in E$ guides the methodology to compute the impact of e_2 by considering the previously calculated posterior as the new prior.

During this iteration we assume as evidence the average price a customer spends on products from a specific category. Assuming products from the same category the proposed model computes the degree of belief on whether a product from the “Offers List” is suitable for purchase given its price. This is shown in equation (19),

$$P(H/E_2 = averageprice) = \frac{P(E_2 = averageprice/H)P(H)}{P(E_2 = averageprice)} \tag{19}$$

4.4.5 The Recommendation Engine

Given a list of products on offer (denoted by *OD*), the recommendation engine shown in Figure 10 is configured to make personalized targeted recommendations based on the unique shopping preferences of each user.

As already mentioned, after the construction of sub-datasets of products for each user belonging to a cluster along with *RK*, the system applies the Bayesian approach given a pair of products from the *RK* with products from the *OD* dataset in order to calculate the posterior probability.

The offers in the *OD* are divided into various categories, like for example, 2 products for the price of 1, products that are on sale, or products that had a reduction in price. When there are no any offers that suit the customer's unique shopping preferences, the system recommends offers that are suitable for the whole group of users (cluster).

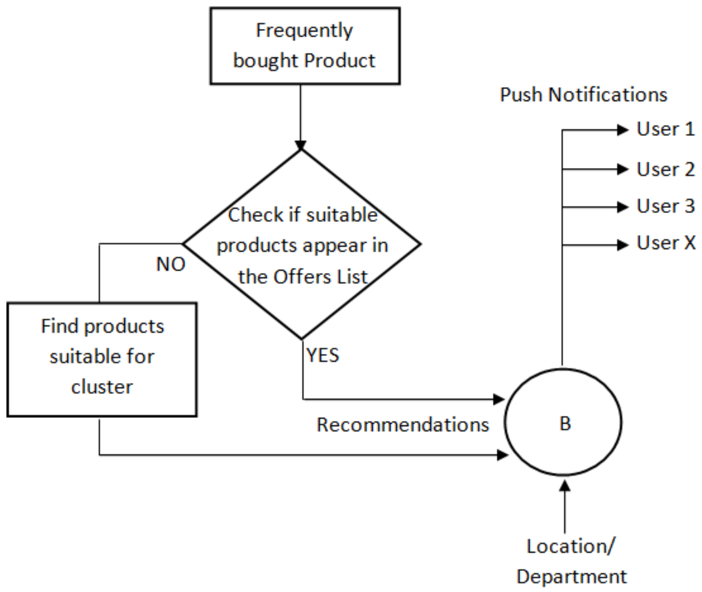

Figure 10: Recommendation Engine

4.4.6 Point-Of-Sale (POS):

To facilitate more the shopping experience of customers in a store, iBeacons are installed close to the POS systems to push notifications to customers about products that have not been purchased but appear in the personalized "To-Buy-List" of each customer as shown in Figure 11. After a successful payment, the recent transaction history is uploaded on a Cloud server. In addition, the system updates the user's dynamic information used for future purchases. Finally, the system checks whether a customer has bought products from the recommendation list. In such a case, it uses that information to derive statistical evidence of any increase on the store's revenue as a result of purchases made from the recommendation list.

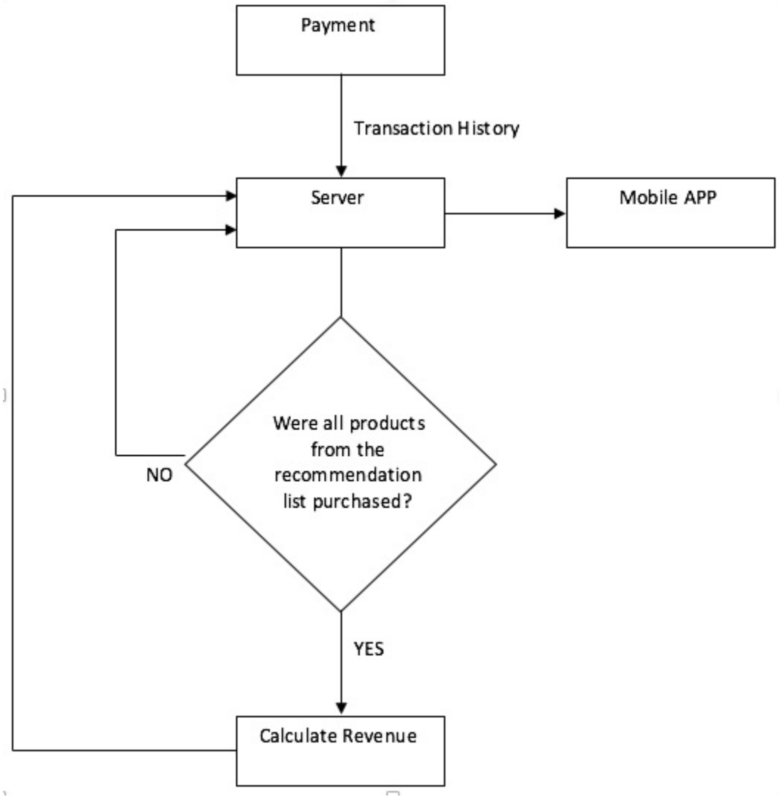

Figure 11: Supermarket POS point

4.4.7 Preliminary Experimentation

This section describes a preliminary experimental case-study that has a twofold purpose: (i) to observe an optimal value for the parameter β used to determine the number of clusters, and (ii) to get a confirmation of the validity of the techniques used.

4.4.7.1 Case-Study: Local Supermarket

For the purposes of our experimentation we deployed our system in a local supermarket with 200 customers (i.e., users). By analysing the data generated from deployment we constructed a set of datasets as follows: (a) Users-Static dataset, defined as SU, that contains the users' static information, such as: Customer ID, Full Name, Health Status, Allergies, Nutrition characteristics etc.; (b) Users-Dynamic dataset, defined as DU, that includes real-time information such as transaction history, to-buy-list and location; (c) Products dataset, defined as DP, which contains information regarding items; (d) Offers dataset, defined as OD; that includes special-offer products; (e) "to-buy-list" dataset, defined as TBD, that contains products that are essential to buy. We asked 200 users to provide us with their static information that was not available from the supermarket's proprietary system. iBeacons were installed in 2 different departments of the supermarket (i.e., Health care and Grocery). The DP dataset contains products from these departments where the missing features for each product were filled in manually. Finally, for evaluation purposes, we asked the users to participate in a short survey where they have been asked to rate their recommendations (top-5 items).

The outcome of the above process resulted in the following datasets: the SU dataset consists of 200 users with 8 features, the DU dataset consisting of 3000 users' historical data monitored from 15 visits for each user between January to June 2016 with 6 features, the DP dataset which contains 1000 products with 14 features, the OD dataset consisting of 45 products that are on offer with 5 features. Finally, the TBD dataset contains 50 products from two departments with 2 features.

The experiments were carried out on a Pentium (R) Dual-Core 2.70 GHz machine with 4GB of main memory running Windows 7 (64 bit).

4.4.7.2 The Parameter β

To determine an appropriate set of clusters for the recommendation process the entropy threshold similarity value β (Stylianou & Andreou, 2007), needs to be tuned on the size of the dataset. The higher the value the lower the number of the clusters discovered as observed in Table 16. The execution time increases proportionally as the number of clusters increases.

Table 16: Impact of value β on the SU dataset

β	**# of cluster**	**Execution time (s)**
1	3	0.37
0.75	4	0.39
0.5	7	0.44
0.25	11	0.61
0.1	23	0.76

By observing the sensitivity of the β parameter over a set of empirical trial-and-error experiments, we set β = 0.5 as the optimal value to obtain a number of clusters in a relatively short execution time. The resulted number of cluster after utilizing the Entropy-based approach and the Hard k-modes clustering was 7. Having determined the set of clusters for the users, a set of rules was applied to create the sub-datasets of products suitable for each cluster.

4.4.7.3 A Demonstration Example Cluster

To demonstrate our approach, let *c1* be a cluster that has the following centroid <1,2,1,1,0,0,0,1> where each element of the feature vector describes the categories depicted in Table 17.

Table 17: Example of features described by the centroid vector

No.	Value	Description	Feature
1st	1	Cyprus	Country
2nd	2	Greek	Nationality
3rd	1	Christian	Religion
4th	1	True	Fasting
5th	0	False	Vegetarian
6th	0	False	Diet
7th	0	False	Allergies
8th	1	Good	Health Status

The feature vector of the centroid is used for applying the priority rules. The output of this filtering process is a sub-dataset of products, $p_j \in PD$, where j denotes the cluster number. Hence, $p_1 \in PD$ is the sub-dataset of products for cluster c_1. Following our demonstration and utilizing Algorithm 1 on the DU dataset, we derived for $user_{12}$ in cluster c_1 the products that the user buys more frequently. The system selects the top frequently bought products for $user_{12}$ on the Shampoos category to be compared with products of the same category belonging to the OD dataset. This is depicted in Table 18.

Table 18: Top frequency bought products

user #	cluster #	cat: Shampoos	f score
$user_{12}$	c_1	Product 1	0.36
		Product 232	0.27
		Product 31	0.20
		Product 4	0.17

Using the Bayesian Inference approach, the system calculates the posterior probability to guide the recommendation process as:

Iteration 1:

Let us assume that the likelihood shown in equation (20),

$$P(E_1 = simScore|H) \tag{20}$$

is given by a probability density function presented in equation (21) as an integral over a finite region

$$[a, b], P(a \leq X \leq b) = \int_a^b f(x)dx \tag{21}$$

Following the experiment of $user_{12}$, the system computes the posterior degree of belief between the similarity score derived using cosine-similarity measure (Ning et al., 2015), by comparing the text descriptions of Product 1 from the Shampoo category with Products $\in$ *OD*. We also assume that *P(H)* is a uniform prior using the principle of indifference.

Iteration 2:

Let us assume that the likelihood for the second evidence that is presented in equation (22),

$$P(E_2 = avgPrice|H) \tag{22}$$

is given by a probability density function as an integral over a finite region as equation above.

Now the previously calculated posterior for $user_{12}$ is updated using a new evidence i.e., the average price that $user_{12}$ spends on products from the Shampoo category. In this iteration, the system computes a new posterior degree of belief given the average price spent on products with products that belong to OD. Note that in order for a product to be recommended it must exist in the dataset of the cluster to which a user belongs to. Following our example with $user_{12}$, if a product does not exist in p_1 it cannot be recommended.

4.4.7.3.1 Recommendations:

The proposed system recommends to $user_{12}$ a list of products that are on offer similar to Product 1 as shown in Table 19.

Table 19: Recommended products for $user_{12}$

product id	probability of purchase
Product 321	0.76
Product 88	0.65
Product 454	0.59
Product 125	0.46
Product 779	0.39

The proposed methodology returns a personalized list of top-5 items that are on offer. The recommendations list is broadcasted to $user_{12}$ using iBeacons that are located in the Health care department.

4.4.7.3.2 Measuring Accuracy

To evaluate how well the proposed methodology recommends products that are on offer to users, we measure precision using the metric presented in equation 23 as follows: *(i)* From the set of 200 users that visited the store 15 times, the system suggests the top-5 products that each user is likely to purchase from the offers list. *(ii)* At checkout the system monitors which of the recommended products were actually purchased. *(iii)* Along with the above data, users are asked to provide explicit feedback by annotating with (Suitable/Not Suitable) which of the suggested top-5 products might be of an interest to them as presented in Table 20.

Table 20: Obtaining feedback from $user_{12}$

product id	probability of purchase	purchased at POS?	suitable?
Product 321	0.76	Yes	N/A
Product 88	0.65	No	No
Product 454	0.59	No	Yes
Product 125	0.46	No	Yes
Product 779	0.39	No	No

The feedback phase seeks to obtain additional information from users that guides our evaluation. The feedback is collected in the form of true positive that is product recommended and purchased or suitable for purchase, and false positive that is product recommended but not purchased and not suitable. From these annotations, the precision is calculated as shown in equation (23),

$$precision = \frac{tp}{tp + fp} \quad (23)$$

For $user_{12}$ during a single visit this value is equal to 0.60.

To monitor the behavior of the system, we repeated the experiment with the set of users that participated in our feedback experiment. Figure 12 shows a frequency plot with the precision (y-axis) obtained for each user grouped into bins. The average precision for this case is 0.7190. Similarly, to study how the number of recommended items affects the system's precision, we repeated the experiment for the top-3 items as shown in Figure 13. The average precision for this case is 0.8028.

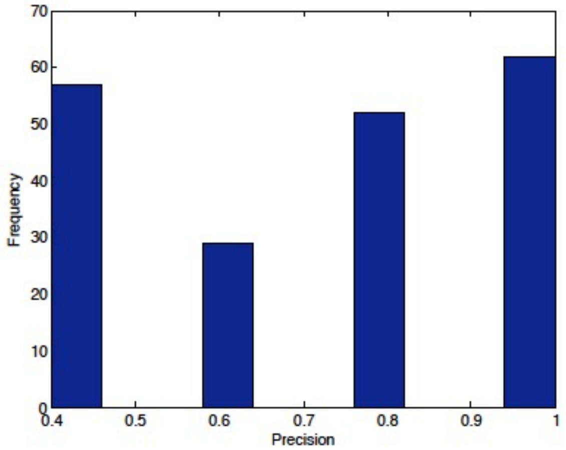

Figure 12: Users' precision for the top-5 case

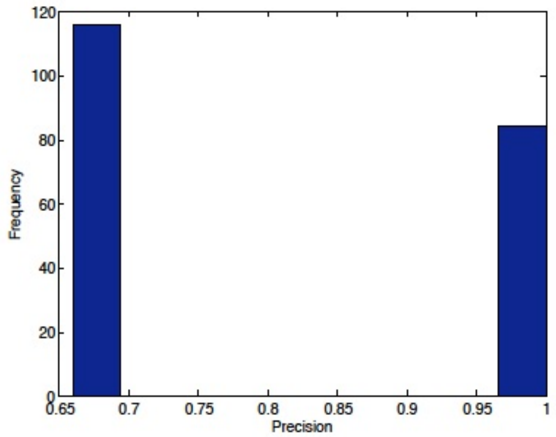

Figure 13: Users' precision for the top-3 case

4.4.7.4 Comparison with benchmark CF algorithms

To compare the accuracy of the proposed methodology with common CF approaches (item-based, user-based) we additionally asked customers to evaluate the lists of recommended products resulting from both item-based and user-based methods. Table 21 shows the precision between the proposed approach compared with the average precision computed from the aforementioned CF approaches. We observed that our system performs better than the typical CF approaches in various cases of recommended products, with an average improvement of 24.7%.

Table 21: Precision comparison with the benchmark CF algorithm

product id	Proposed model	Average benchmark CF
top-3	0.8028	0.6243
top-5	0.7190	0.5951

4.4.7.5 Overview

In a supermarket setting, where users are constantly changing their shopping preferences or habits, and products change their characteristics or lose their popularity, there is a need for a system that captures the dynamic environment of a supermarket aiming to recommend products that are on offer. Throughout this work we discussed how the cold-start problem, the data sparsity and other scalability issues often met in RSs are minimized by utilizing clustering methodologies. Moreover, we present how the system can suggest personalized recommendations to users considering different pieces of evidence with high accuracy when utilizing a Bayesian inference approach. Finally, the proposed approach was implemented in a real-world scenario and compared with other methods; the final results indicated that the proposed methodology performs better than the traditional CF approaches (item based and user-based) in terms of accuracy.

The work outlined in this section lead to the publication of one a conference paper that deployed a RS to produce real-time personalized offers was published (Christodoulou et al., 2017).

Chapter 5: Improving the Performance of Classification Models

5.1 Introduction

The prediction process applied in every scientific discipline is considered as highly complex exhibiting high levels of uncertainty as it involves multiple and usually conflicting factors. Therefore, the prediction problem is particularly challenging and researchers aim to find various solutions on how to produce predictions (recommendations) with high accuracy in a short amount of time.

This chapter introduces various hybrid prediction models that exploit the advantages offered by Fuzzy Cognitive Maps (FCMs) coupled with the prediction abilities of classification models such us Support Vector Machines (SVM), Linear Discrimination (LDA), k-NN and Classification Trees to tackle the aforementioned challenges, as well as the cold-start problem and scalability issues that exist in RS, in order to produce more accurate results.

The proposed models first use a FCM to discover correlation patterns and interrelationships that exist between the data variables and form a single latent variable. This variable is then inserted into the classification models both during the training and testing phases, to improve prediction capabilities. The efficacy of the hybrid models is demonstrated through its application on two different domains. The proposed models are evaluated on an occupancy dataset and then on a medical dataset aiming firstly to deal with the prediction accuracy problem and then produce recommendations to tackle the RS challenges. Experimental results show that the hybrid models perform better than traditional models.

5.2 Related Work

This section starts by briefly presenting other studies that use the datasets utilized in this work and then focuses on the use of the underlying models in prediction problems.

The accuracy of predictions that depends on occupancy using various data attributes (light, temperature, humidity and CO2) was first presented in Candanedo & Feldheim, (2016). This work uses three datasets, one for training the models and two for testing them. A number of training models such us the Linear Discriminator Analysis (LDA), Regressions Trees and Random Forests were used for training and testing purposes. The best accuracy obtained from the several experiments ranges from 95% to 99%. Results showed that the impact of accuracy on each experiment depends on the classification model and the number of features selected each time. Taking into consideration all of the features, the best accuracy was yielded using the LDA model for both test datasets.

The work described in Smith et al., (1988) uses a neural network to predict the diabetes mellitus for a high risk population in India. It was one of the first algorithms used in health forecasting. The proposed methodology was compared with other models achieving a high accuracy of 76%.

Ster et al. (1996) test a number of classification systems on various medical datasets (Diabetes, Breast Cancer and Hepatitis) in order to obtain accurate results when using a number of different methods. In terms of classification accuracy, in most of the datasets the neural networks approaches outperform other methods such as Linear Discrimination Analysis (LDA), K-nearest neighbor, Decision Trees and Naïve Bayes.

In Papageorgiou et al., (2016), a new hybrid approach based on FCM and ANN is presented for dealing with time series prediction. The proposed model was applied and tested in predicting water demand on the island of Skiathos, Greece. The methodology presented increases prediction accuracy of ANN by using concepts from FCMs as input data.

The authors in (Papageorgiou, & Poczeta, 2015) conducted a multivariate analysis and forecast of the electricity consumption with a 15-minute sampling rate using three different FCM learning approaches: multi-step gradient method, RCGA and SOGA. These approaches were found to be more suitable for the electricity consumption prediction rather than popular artificial intelligent methods of ANNs and ANFIS.

The authors in (Shin et al., 2005) investigate the application of a SVM model to a bankruptcy prediction problem. Even though it is known from previous studies that the back-propagation neural network (BPN) produces accurate results when dealing with pattern recognition tasks, it faces limitations on constructing an appropriate model for real-time predictions. The proposed classification model based on SVM captures the characteristics of a feature space and is able to find optimal solutions using small sets of data. The suggested approach performs better than the BPN in terms of accuracy and performance when the training size decreases.

The work presented in (Mohandes et al., 2004) introduces a SVM model on a wind speed prediction problem. The performance of the proposed methodology was compared with a multilayer neural network (MLP). The dataset used for experimental purposes was recorded in Asia and the results based on the RMSE error between the actual and predicted data showed that the SVM approach outperforms the MLP model.

Cortes & Vapnik, (1995) explore different machine learning techniques in order to predict the burnt area of forests. Two models, SVM and Random Forests, were tested offline on a real-world dataset collected from a region in Portugal. On each experiment the two models make use of various features and their accuracies are computed. The best approach used the SVM algorithm with all meteorological data as input and was able to predict the burnt area of small fires which happen more frequently. A drawback of this approach is that it cannot predict with high accuracy the burnt area of larger fires; this is feasible only by adding additional information to the model.

Finally, a similar approach is followed in (Papageorgiou et al., 2006) that presents a Fuzzy Cognitive Map (FCM) trained using a Nonlinear Hebbian Algorithm combined with Support Vector Machines (SVMs) in order to address the tumor malignancy classification problem by making use of histopathological characteristics. The hybrid model achieves a classification accuracy of 89.13% for high grade tumors and 85.54% for low grade tumors and outperforms on overall accuracy the k-nearest neighbor, linear and quadratic classifiers. Nevertheless, this methodology uses a SVM approach to classify the data and not to train the model.

In this section our intention was to show indicative examples of problems that these models may tackle, their prediction strengths and abilities, and the diversity of the application domains that may be benefitted by them so as to provide a form of justification for their selection as constituents of our hybrid model.

5.3 Overview of approach

This section presents the approach for developing a FCM model to discover hidden correlations that exist in the training set and subsequently integrating these correlations into classification models aiming to increase the model's accuracy.

5.3.1 Fuzzy Cognitive Maps

Fuzzy Cognitive Maps (FCMs) are tools which inherit elements from the theory of fuzzy logic and neural networks (Kosko, 1993). The FCM approach was firstly proposed by Kosko as an extension of Cognitive Maps (Kosko, 1986) and was firstly used as decision support tools in various scientific fields, such as social and political developments, urban planning, agriculture, information and communication technology, software engineering and others (Kosko, 2010). Their simple nature and ease of understanding led them to be used in a wide range of applications. Essentially, a FCM is a digraph with nodes representing concepts in the domain of a problem and directed edges describing the causal relationships between those concepts. A positively weighted directed edge between two concepts indicates a strong positive correlation between the causing and the influenced concept. Inversely, a negatively weighted directed edge indicates the existence of a negative causal relationship. Two conceptual nodes without a direct link are, obviously, independent. Each concept node keeps a numerical value as an activation level in the range [0, 1], and indicates the strength of its presence in the problem under study. The number of nodes and the number of their causal relationships denote the degree of complexity of the map. Additional complexity appears with the presence of cycles between nodes, that is, paths starting and ending on the same nodes.

As originally proposed, a FCM is constructed with the aid of a group of experts who, based on their knowledge and expertise, identify the nodes that are relevant to the problem under study and define the activation levels of the concepts, as well as the weights of the causal relations between them. The model is then executed on a series of discrete steps (Kosko, 1986) during which the activation levels of the participating concepts are iteratively calculated for a number of repetitions. At the end of the execution cycle the model can either reach an equilibrium state at a fixed point, with the activation levels reaching stable numerical values, or exhibit a limit cycle behavior, with the activation levels falling in a loop of numerical values under a specific time-period, or present a chaotic behavior, with the activation levels reaching a variety of numerical values in a random way. In the former two cases inference is possible.

The activation level of a node denotes its presence in the conceptual domain and is calculated taking into account the activation levels of the nodes from which it is fed, as well as its own current activation. The activation level of each node is calculated using equation (24).

$$x_i^{t+1} = f\left(\sum_{j \neq i} w_{ij} x_i^t + x_j^t\right) \tag{24}$$

where *f* is a threshold function that keeps an activation level value in the desired interval and it can be chosen from a number of available functions (Bueno & Salmeron, 2009) based on the nature of the model and the problem in hand. The sigmoid function shown in equation (25) is the most widely used function and squashes the value of the function in the interval [0, 1]:

$$f(x) = \frac{1}{1 + e^{=\lambda x}}, \lambda > 0 \tag{25}$$

In this work the FCM model is used to discover the latent variable FCM_{OUT} that will be later used as input to the classification models.

5.3.2 FCM Construction

A semi-automated learning method for the FCM construction is proposed based on the correlations between the input variables calculated using historical data in combination with literature review on the topic and domain expert's consultation. The proposed approach follows a stepwise process described in details below:

Step 1: Pre-processing

At this step the historical data is fed using a pre-processing procedure during which a linear normalization is performed as shown in equation (26) that transforms the input data set values in the range [0, 1].

$$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}}, i = 1..n \quad (26)$$

Step 2: Correlation Matrix Calculation

A strong indication of the dependence between the input variables is given by calculating their correlation and associated *p*-values.

Step 3: Literature Review & Expert Consultation

The connections between the concepts in a FCM represent the one-way causality from one to another. By default, correlation is not causality, i.e. we cannot safely argue about the source and destination of causalities between two correlated nodes; thus, we need to examine this issue further. We resort to use domain experts and/or knowledge embodied in the relevant literature to accept or discard possible causalities as these are extracted from the correlation matrix and then decide upon the direction of each causality. In addition, we make some valid assumptions that come logically and effortlessly regarding variables that depend on time such as day, week etc. which cannot be influenced by any other variable.

Step 4: FCM Analysis & Calibration

A significant factor that greatly affects FCM performance is the selection of the activation level equation, as well as the selection of the threshold function. The criteria for this decision may be attributed to the map's balance based on negative and positive cycles and input and output data format (Andreou et al., 2005).

The performance evaluation of a FCM can be made by assessing the success rate of the model over training data, aiming to reach the maximum possible level. Based on the type of real reference values we can define the form of the model's output, $FCMout_i$ where i represents each object in a dataset. For example, if the reference values' class is binary, we may seek for a threshold that could separate $FCMout_i$ values in such a way so as to deliver the maximum matching between the predicted class $FCMout_i$, and the actual x_i values as shown in equation (27):

$$max\{|x_i = FCMout_i|\}, i = 1..n$$

$$FCMout_i = 1, if FCMout_i > threshold \quad (27)$$

$$FCMout_i = 0, otherwise$$

In the case of a scalar reference value type a regression analysis, as presented in equation (28), can be applied using the *FCMout* values as the dependent variable $\boldsymbol{x}$ and the reference values as the explanatory variable *y*:

$$y_i = \alpha x_i + \beta \quad (28)$$

Steps 2, 3 and 4 of the FCM analysis and calibration process may be repeated leading to the construction of the final FCM model as depicted graphically in Figure 14.

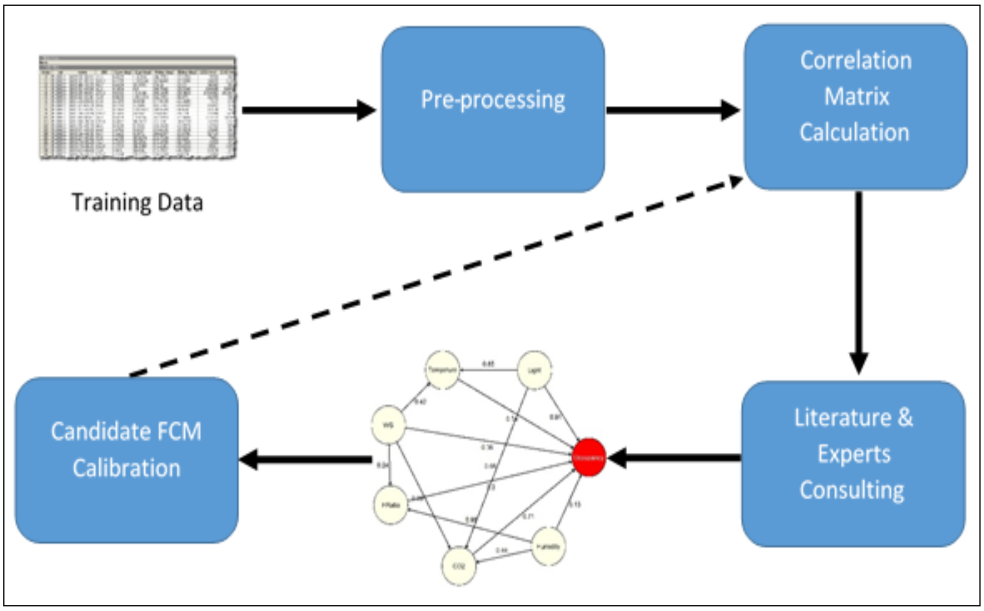

Figure 14: FCM model constructions

5.3.3 Datasets & Modelling

To evaluate the performance of our methodology we used two datasets from different domains. This section provides an overview on those datasets.

5.3.3.1 Occupancy Dataset

The occupancy dataset used in this work was first presented in (Candanedo & Feldheim, 2016). The data samples were collected from sensors installed in an office room, while a digital camera was used to find out the occupancy of the room.

In this research we utilize a training dataset as well a small and a large dataset for testing purposes. Each dataset has the following attributes: Date and time in the format of year-month-day hour:minute:second, Temperature measured in Celsius (*T*), Relative Humidity in % (φ), Light measured in Lux (*L*), CO2 in ppm (*CO2*) and the Humidity Ratio (*W*) that is calculated by dividing the temperature and relative humidity. Occupancy (O) is either 0 for not occupied or 1 for occupied status. The utilization of date-time stamp in all datasets has been used to extract two additional variables: Number of seconds since midnight (*SSM*) and week status (WS) that is either 0 for weekend or 1 for weekday. The training dataset consists of 8143 records with 7 attributes (6 attributes from the original dataset plus the attribute resulted in by the FCM) and the two test datasets consist of 2665 and 9572 records respectively with 6 attributes (5 attributes from the original datasets plus the FCM attribute).

The aforementioned datasets were used to train and test the proposed model and the baseline algorithms presented in the next sections for comparison purposes. The datasets used by the proposed approach also include the variable discovered by the Fuzzy Cognitive Map. The occupancy attribute of the test datasets was used for evaluating and comparing the proposed approach against the baseline methods.

5.3.3.1.1 Modelling

Following the model construction procedure described above, we proceeded and utilized the available dataset to construct and evaluate the proposed model.

Firstly, a linear normalization was applied to the datasets, as a result of which values in both the training and test datasets were transformed in the range [0, 1]. Subsequently, the correlation matrix was calculated and it is presented in Table 22 with *p*-values appearing in parentheses.

Based on the findings extracted from the correlation and associate *p*-values, we identified pairs with high linear significant relationships. In addition, to support this procedure, we defined and set some rules towards causality identification. We eliminated the SSM variable since its values do not have a continuous and linear relationship with the Occupancy variable. We also inferred that the time depended variable WS cannot be influenced by any other variable. Finally, we consulted the relevant literature, where necessary, to identify the value and the direction of an influence.

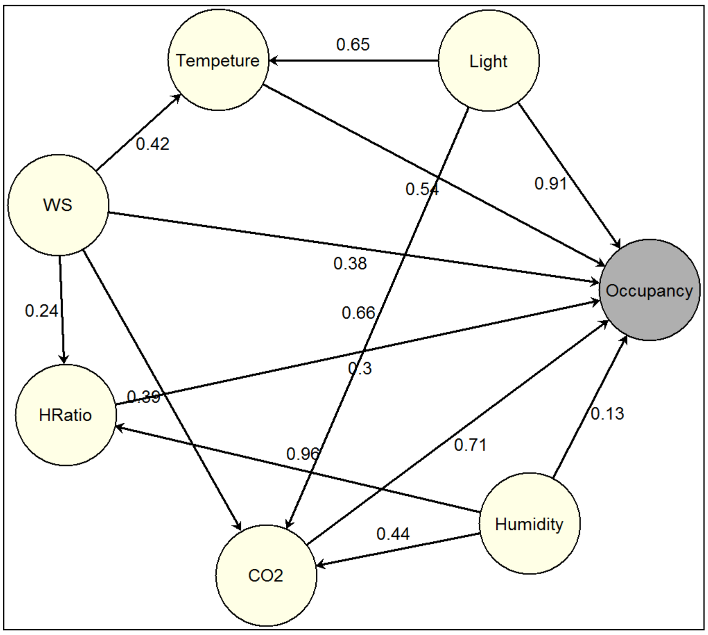

Figure 15: Final FCM model for the occupancy dataset

The FCM model that emerged from this process is shown in Figure 15 with its associated edge-weight values. It is obvious that the constructed FCM is a fully positive map consisting exclusively of positive cycles. This evidence guided us to the selection of the update and threshold functions in such a way so as to increase the model's sensitivity to uncertainty and to quantization of the final value.

Table 22: Correlation Matrix and *p-values* in parentheses for the occupancy dataset

	WS	**SSM**	**T**	**φ**	**L**	**CO2**	**W**	**O**
WS	0.00	-0.01	0.42	0.11	0.28	0.39	0.24	0.38
		(0.33)	*(0.00)*	*(0.00)*	*(0.00)*	*(0.00)*	*(0.00)*	*(0.00)*
SSM	-0.01	0.00	0.26	0.02	0.09	0.21	0.10	0.08
	(0.33)		*(0.00)*	*(0.00)*	*(0.00)*	*(0.00)*	*(0.00)*	*(0.00)*
T	0.42	0.26	0.00	-0.14	0.65	0.56	0.15	0.54
	(0.00)	*(0.00)*		*(0.13)*	*(0.00)*	*(0.00)*	*(0.00)*	*(0.00)*
φ	0.11	0.02	-0.14	0.00	0.04	0.44	0.96	0.13
	(0.00)	*(0.13)*	*(0.00)*		*(0.00)*	*(0.00)*	*(0.00)*	*(0.00)*
L	0.28	0.09	0.65	0.04	0.00	0.66	0.23	0.91
	(0.00)	*(0.00)*	*(0.00)*	*(0.00)*		*(0.00)*	*(0.00)*	*(0.00)*
CO2	0.39	0.21	0.56	0.44	0.66	0.00	0.63	0.71
	(0.00)	*(0.00)*	*(0.00)*	*(0.00)*	*(0.00)*		*(0.00)*	*(0.00)*

W	0.24 *(0.00)*	0.10	0.15 *(0.00)*	0.96 *(0.00)*	0.23 *(0.00)*	0.63 *(0.00)*	0.00	0.30 *(0.00)*
O	0.38 *(0.00)*	0.08 *(0.00)*	0.54 *(0.00)*	0.13 *(0.00)*	0.91 *(0.00)*	0.71 *(0.00)*	0.30 *(0.00)*	0.00

The sigmoid function with $\lambda=1$, was chosen as the threshold function and the update function is described in equation (29) (Iakovidis, & Papageorgiou, 2011):

$$x_i^{t+1} = f\left(\sum_{j\neq i} w_{ij}(2x_i^t - 1) + (2x_j^t - 1\right) \quad (29)$$

We performed a number of executions on the FCM model to check its accuracy performance. For each execution we ran 50 iterations and managed to reach equilibrium in all executions and to deliver a stable value for *FCMout*. The binary type of the reference occupancy value (0 or 1) allowed us to use a threshold value equal to 0.27 achieving 98.8% accuracy on the training input data. After the finalization of the FCM model the hybrid models were executed on the two occupancy test datasets.

5.3.3.2 Diabetes Dataset

For further examining the performance of our approach against baselines we utilize a second dataset. The “Pima Indian Diabetes” dataset first presented in (Smith et al., 1988) was also used for evaluation purposes. 768 cases have been collected in which 500 were healthy and 268 with diabetes.

The dataset presents eight different attributes that describe the age (*age*) in years, number of times pregnant (*p*), body mass index (*bmi*) expressed as (weight in kg)/(height in m)^2, plasma glucose concentration (*g*) in mg/dl, triceps skin fold thickness (*st*) in mm, diastolic blood pressure (*bp*) in mm Hg, diabetes pedigree function (*dpf*), 2-hour serum insulin (*i*) in U/ml and finally the diabetes outcome class variable (*out*) which can be either 0 or 1. Following the same reasoning as described in (Candanedo & Feldheim, 2016), we used 576 rows from the dataset as training inputs and the rest 192 as test inputs. The training dataset, as well the test one, include also the variable discovered by the Fuzzy Cognitive Map.

5.3.3.2.1 Modelling

Using the same rationale as with the previous experimental case (Occupancy dataset), we proceeded and utilized the diabetes dataset to construct and evaluate the proposed model. After the application of linear normalization to the dataset, the correlation values extracted along with their associated *p-values* are presented in Table 23. Moreover, we employed two diabetologists as domain experts in order to identify and confirm causalities. The FCM model that emerged from this process is shown in Figure 16 with the associated edge-weight values.

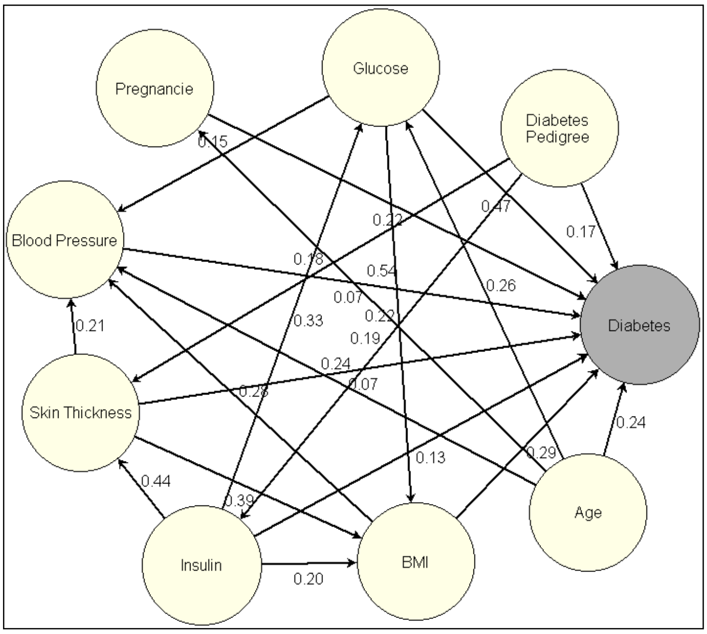

Figure 16: Final FCM model for the diabetes dataset

Table 23: Correlation Matrix and *p-values* in parentheses for the diabetes dataset

	Pregnancy	**Glucose**	**Blood pressure**	**Skin thickness**	**Insulin**	**BMI**	**Diabetes Pedigree**	**Age**	**O**
Pregnancy	0.00	0.129 *(0.0003)*	0.141 *(8x10^{-5})*	-0.081 *(0.023)*	-0.073 *(0.041)*	0.017 *(0.624)*	-0.033 *(0.353)*	0.544 *(1x10^{-60})*	0.221 *(5x10^{-10})*
Glucose	0.129 *(0.0003)*	0.00	0.152 *(2x10^{-5})*	0.057 *(0.112)*	0.331 *(3x10^{-21})*	0.221 *(5x10^{-10})*	0.137 *(0.0001)*	0.263 *(1x10^{-13})*	0.466 *(8x10^{-46})*
Blood pressure	0.141 *(8x10^{-5})*	0.152 *(2x10^{-5})*	0.00	0.207 *(6x10^{-9})*	0.088 *(0.013)*	0.281 *(1x10^{-15})*	0.041 *(0.253)*	0.239 *(1x10^{-11})*	0.065 *(0.071)*

Skin thickness	-0.081 *(0.023)*	0.057 *(0.112)*	0.207 *($6x10^{-9}$)*	0.00	0.436 *($3x10^{-37}$)*	0.392 *($1x10^{-29}$)*	0.183 *($2x10^{-7}$)*	-0.113 *(0.001)*	0.074 *(0.038)*
Insulin	-0.073 *(0.041)*	0.331 *($3x10^{-21}$)*	0.088 *(0.013)*	0.436 *($4x10^{-37}$)*	0.00	0.197 *($3x10^{-8}$)*	0.185 *($2x10^{-7}$)*	-0.042 *(0.243)*	0.130 *(0.0002)*
BMI	0.017 *(0.624)*	0.221 *($5x10^{-10}$)*	0.281 *($1x10^{-15}$)*	0.392 *($1x10^{-29}$)*	0.197 *($3x10^{-8}$)*	0.00	0.140 *($9x10^{-5}$)*	0.036 *(0.315)*	0.292 *($1x10^{-16}$)*
Diabetes Pedigree	-0.033 *(0.353)*	0.137 *(0.001^{3})*	0.041 *(0.253)*	0.183 *($2x10^{-7}$)*	0.185 *($2x10^{-7}$)*	0.140 *($9x10^{-5}$)*	0.00	0.033 *(0.352)*	0.173 *($1x10^{-6}$)*
Age	0.544 *($1x10^{-60}$)*	0.263 *($1x10^{-13}$)*	0.239 *($1x10^{-11}$)*	-0.113 *(0.0015)*	-0.042 *(0.243)*	0.036 *(0.315)*	0.033 *(0.352)*	0.00	0.238 *($2x10^{-11}$)*
O	0.221 *($5x10^{-10}$)*	0.466 *($8x10^{-43}$)*	0.065 *(0.071)*	0.074 *(0.038)*	0.130 *(0.0002)*	0.292 *($1x10^{-16}$)*	0.173 *($1x10^{-6}$)*	0.238 *($2x10^{-11}$)*	0.00

Similarly to the occupancy model, the constructed FCM for the diabetes case is a fully positive map consisting exclusively of positive cycles. For this dataset we selected also the sigmoid function with $\lambda=1$, as the threshold function.

A number of executions were performed on the constructed model to assess its accuracy performance. 50 iterations ran on each execution and the model managed to reach equilibrium in all cases delivering a stable FCM_{OUT} value. This value was again utilized by the classification model.

5.3.4 Machine Learning Classification Models

5.3.4.1 Support Vector Machines

SVM is a supervised machine learning approach that analyzes data for classification. It constructs a model and assigns instances in different categories. Instances are represented as points in the feature space and they are divided by a hyperplane (Cortes & Vapnik, 1995). When new data is available it is mapped onto the space and the model predicts the specific category it belongs to based on the side of the hyperplane it falls on.

The aim of a SVM model is to find and select the best hyperplanes to separate the data. This work utilises a Linear SVM algorithm that consists of the following steps:

Step 1: Hyperplanes

Given a hyperplane H_o that separates D and satisfies equation (30):

$$w \cdot x + b = 0 \tag{30}$$

where w is a weight and b is a threshold. Select two other hyperplanes H_1 and H_2 that also separate the data shown in equation (31) and (32):

$$w \cdot x + b = \delta \tag{31}$$

$$w \cdot x + b = -\delta \tag{32}$$

where δ is a variable, so that the distance of H_o from H_1 and H_2 is equal.

Each vector x_i can belong to a class if one of the following equations (33) and (34) is satisfied:

$$w \cdot x_i + b \geq 1 \tag{33}$$

$$w \cdot x_i + b \leq -1 \tag{34}$$

Combining both equations above we get a unique constraint as shown in equation (35) where there are no points between the two hyperplanes:

$$y_i(w \cdot x_i + b) \geq 1 \tag{35}$$

for all $1 \leq i \leq n$ where x_i is the i^{th} training sample and y_i is the correct output.

Step 2: Margin

The hyperplane that has the largest margin between the two classes is used as the best choice to classify the data.

The margin is calculated using equation (36):

$$m = \frac{2}{w} \tag{36}$$

To calculate the optimal hyperplane, the SVM finds the couple *(w, b)* for which w is minimized as presented in equation (37):

$$y_i(w \cdot x_i + b) \geq 1 \tag{37}$$

where x_i is the i^{th} training sample and y_i is the correct output.

Step 3: Classification

The algorithm is trained to find the best hyperplane using the previous steps; it then uses the test data to predict the specific class each sample belongs to.

5.3.4.2 Weighted k-NN

The *k-NN* is one of the most well-known approaches used for classification. This algorithm firstly finds a number of k nearest neighbours for each instance by measuring a distance using various metrics and then it uses that metric to classify each instance taking into consideration the majority label of its nearest neighbours (Gou et al., 2012). This work uses a variation of the traditional *k-NN* approach called Weighted *k-NN* which provides a higher weight to closer neighbours and better accuracy than the normal *k-NN*.

5.3.4.3 Linear Discrimination Analysis

Linear discrimination analysis (LDA) is an approach used in statistics and machine learning in order to find a linear combination of features that separates two or more classes of instances (McLachlan, 2004). LDA follows three steps:

Step 1:

LDA separates the instances $\boldsymbol{x_i}$ given in the dataset $\boldsymbol{D}$ into various groups based on the value of their class. Then it computes the μ value of each dataset and the global μ value of the entire dataset and subtracts those values from the original ones.

Step 2:

The covariance matrix of each group is found and then the pool covariance matrix is calculated using equation (38):

$$C = \sum_{i=1}^{k} p_i c_i(r,s) \tag{38}$$

where c_i is the covariance matrix of group *i, (r,s)* is each entry in the matrix and *p* is the prior probability computed using equation (39):

$$p = \frac{n_i}{N} \tag{39}$$

where n_i defines the total samples of each group and *N* the total samples of the dataset.

Step 3:

At this step the inverse matrix C^{-1} is calculated and used in the discriminant function f_i as shown in equation (40). The discriminant function is the classification rule assinging each object in a class.

$$f_i = \mu_\iota C^{-1} x_i^T - \frac{1}{2}\mu_\iota C^{-1}\mu_i^T + \ln{(p_i)} \tag{40}$$

5.3.4.4 Classification Tree

Classification tree (CT) learning is a commonly used method in machine learning that constructs a tree-like model aiming to predict the class of an instance based on the input of a training dataset (Loh, 2011). In a classification tree model the leaves present the class of an instance and the branches describe the set of features that lead to a leaf (class); therefore, following the decisions from the beginning of the decision tree down to the leaves the classes are predicted.

5.3.4.5 Parameters

This section provides a brief summary of the main parameters used for executing the baseline algorithms and the proposed model.

First of all, a *k-fold* (*k*=5) cross validation was performed on all training models. The Weighted *k-NN* model takes into consideration the 10 closest neighbours (*k*=10), uses the Euclidean distance metric to measure the similarity between instances and the weight is measured using equation (41):

$$\frac{1}{distance^2} \quad (41)$$

The LDA model firstly uses the Baye's Theorem to calculate probabilities; the classifier is constructed based on a linear combination of the dataset's input where the delta threshold is set to 0 and the gamma regularization parameter to 1.

The CT model uses a continuous type of cut at each node in the tree and the Gini's diversity index (gdi) criterion for choosing a split. The SVM model uses a linear kernel function to calculate the classification score of instances and a gradient descent for minimizing the objective function.

5.3.4.6 Accuracy

The accuracy metric used for computing the accuracy and evaluating the methods on both test datasets is presented in equation (42):

$$Accuracy = \frac{A + D}{A + B + C + D} \quad (42)$$

where the A, B, C and D variables are defined in the confusion matrix shown in Table 24.

Table 24: Confusion Matrix

Approach	Predicted Value=1	Predicted Value=0
Reference Value=1	A	B
Reference Value=0	C	D

5.4 Support Vector Machines with Fuzzy Cognitive Maps

This part describes our first attempt to verify our approach by capturing the tendency of the input dataset and deliver a linear output "aligned" with the real predicted value. The construction of the hybrid model presented in Figure 17 comprises two sequential steps: The creation and execution of the FCM model that utilises the available dataset to discover the latent variable *FCMout*; the use of *FCMout* as input, along with the rest inputs of the same dataset, by the SVM model and generation of predictions.

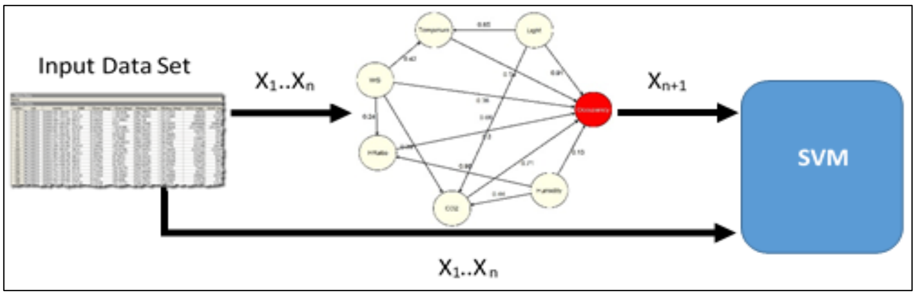

Figure 17: SVM-FCM model ecample

5.4.1 Evaluation

Aiming to evaluate the performance of the proposed approach, we applied the SVM-FCM model on the two different datasets described earlier.

5.4.1.1 Execution Results and Comparison Using the Occupancy Dataset

Table 25 presents the accuracy of the predictions for the baseline approaches, as well the accuracy of the proposed hybrid SVM-FCM model. The *k-NN,* LDA and CT approaches perform well mostly on small datasets and their prediction accuracy declines when the training data size increases. As it can be clearly seen in Table 25 when dealing with a small dataset, the proposed SVM-FCM model has exactly the same high accuracy as the *k-NN* method, it is slightly better than the classic Linear SVM model and the CT model, and presents higher accuracy compared to the LDA approach.

Table 25: Prediction Accuracy

Approach	Test Dataset 1	Test Dataset 2
Weighted *k*-NN	0.9790	0.9601
LDA	0.9674	0.9520
Linear SVM	0.9782	0.9937
Classification Tree	0.9764	0.9726
SVM-FCM	0.9790	0.9945

In the case of the larger dataset our approach clearly outperforms the *k-NN,* LDA and CT methods by an average of 4%. When compared with the classic Linear SVM, the suggested hybrid model again performs slightly better.

5.4.1.2 Execution Results and Comparison Using the Diabetes Dataset

Table 26 presents the accuracy of the proposed methodology compared to the baseline approaches. As it can be seen, the best performing algorithms from the baseline approaches are the Linear SVM and the LDA that score 77.6%. The hybrid model SVM-FCM scores again the higher accuracy with 78.65%, which, when compared against the *k-NN,* LDA, Linear SVM and CT methods, yields an average improvement of 2%.

Table 26: Prediction Accuracy

Approach	Test Dataset
Weighted *k*-NN	0.7344
LDA	0.7760
Linear SVM	0.7760
Classification Tree	0.7448
SVM-FCM	0.7865

5.5 Classification Models with Fuzzy Cognitive Maps

This section continues to explore the prediction performance enhancement of known approaches by FMCs. In this case we capture the main capabilities of FCMs and then integrate the outcome in various classification models to improve their accuracy.

5.5.1 Evaluation

In order to evaluate the performance and accuracy of the proposed methodology on various classification models we follow the same procedure like the *SVM-FCM* case. We refer to the hybrid models that include the *FCMout* input as *SVM-FCM, LDA-FCM, CT-FCM* and *k-NN-FCM*.

5.5.1.1 Execution Results and Comparison of the Occupancy Dataset

The accuracy achieved by the classic classification models is presented in Table 27. Recall that test Dataset 1 refers to the small test dataset of the occupancy dataset while Test Dataset 2 refers to the larger one.

It is easily discernible that the weighted k-NN model outperforms the other approaches when dealing with small datasets. It is also clear that when evaluating the classification models with largest datasets only the Linear SVM method retains high accuracy outperforming the other methods which exhibit a decrease in their predictive ability.

Table 27: Prediction Accuracy of classic models

Approach	*Test Dataset 1*	*Test Dataset 2*
k-NN	0.9790	0.9601
LDA	0.9674	0.9520
SVM	0.9782	0.9937
CT	0.9764	0.9726

Table 28 lists the performance of the hybrid models when executed on both test datasets. It is evident that using the smaller dataset the hybrid models slightly increase the accuracy of the predictions in most cases compared to the classic models except the Classification Tree.

Table 28: Prediction Accuracy of Hybrid Models

Approach	*Test Dataset 1*	*Test Dataset 2*
k-NN - FCM	0.9797	0.9785
LDA - FCM	0.9704	0.9676
SVM - FCM	0.9790	0.9945
CT - FCM	0.9760	0.9764

In the case of the larger dataset the increase of accuracy is observed for all hybrid models. It should also be noted that there is a small decrease on the performance of the hybrid models of k-NN and LDA when executed on the larger dataset compared to the same hybrid models in the smaller dataset, but this decrease is smaller than the percentage of decrease in the case of the classic models. Also, the hybrid model of CT increases its prediction accuracy for the larger dataset compared with that of the non-hybrid model. The linear SVM-FCM model yields the best accuracy on predictions when compared with all other models in the larger dataset.

5.5.1.2 Execution Results and Comparison of the Diabetes dataset

Table 29 presents the accuracy of the classic classification models. As one can easily observe, the best performing methods are the Linear SVM and the LDA that score 77.6%.

The performances of the hybrid models are presented in Table 30. As with the previous experiment, the accuracy levels of most of the models increase except in the case of the CT. The best performing model is the hybrid SVM-FCM.

Table 29: Prediction Accuracy of classic models

Approach	*Test Dataset*
k-NN	0.7344
LDA	0.7760
SVM	0.7760
CT	0.7448

Table 30: Prediction Accuracy of Hybrid models

Approach	*Test Dataset*
k-NN - FCM	0.7552
LDA - FCM	0.7813
SVM - FCM	0.7865
CT - FCM	0.7448

5.5.2 Recommendation Engine

Figure 18 presents a general framework for generating recommendations for newly added data such as users and items. This framework consists of the following steps:

1. The hybrid models are trained using the training data
2. A class is assigned to each data on the test set automatically
3. According to that class the recommendation engine generates recommendations

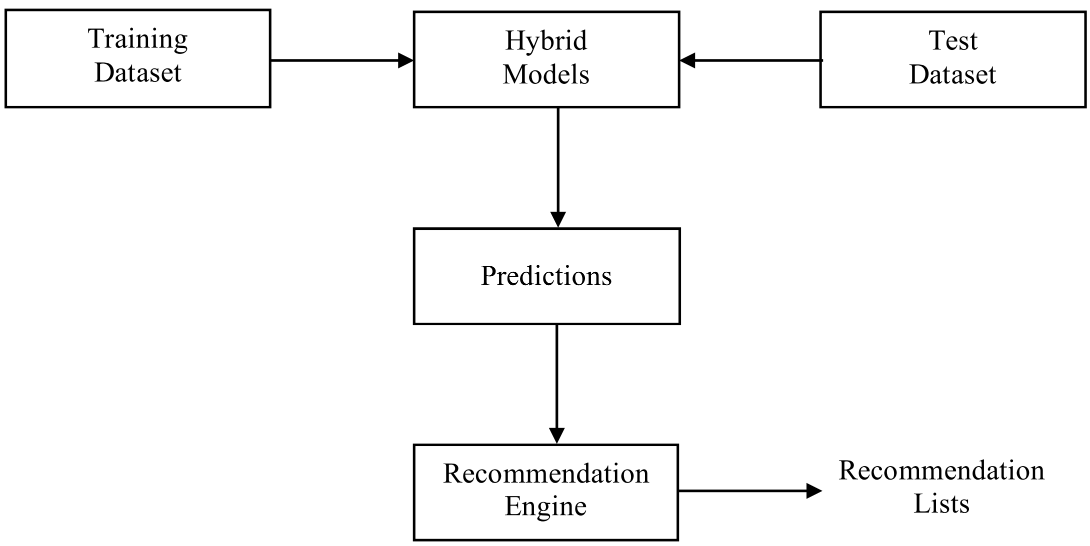

Figure 18: Recommendation process framework

In the following we present how the recommendation engine can be used in a wide range of applications in order to produce recommendations for specific groups of users.

5.5.2.1 Occuppancy Dataset

As described in the previous section hybrid models can predict with high accuracy if a room is occuppied or not. In the case that a room is occupied the recommendation engine presented in this section can suggest various solutions to users to reduce the energy consumption. The recommendation list consists of :

- Shutdown devices (projector, computers) if not needed
- Choose the right LEDs
- Unplug chargers
- Use a power strip
- Turn off lights

5.5.2.2 Diabetes Dataset

Following the same procedure as described above, when a person has a diabetes the recomendation process produces a list of suggestions that can help a person to deal with diabetes and live a normal life. The recommendations include the following suggestions:

- About diabetes types
- Diabetes ABCs (A1C, Blood pressure, and Cholesterol)
- How to live with diabetes (stress, exercise, diet)
- Routine care check

5.5.2.3 Shop Dataset

Hybrid models can identify based on historical data if a customer belongs to the about-to-leave group. In such a case the recommendation engine tries to suggest solutions to them in order to prevent them from leaving such as:

- Discounts
- Rewards
- Coupons
- Special offers
- Exclusive products

5.5.3 Overview

This part introduced a series of hybrid models that combine FCM and classification models to improve the prediction accuracy of the classic models and at the same time to tackle the cold-start, scalability and other issues that exist in RS. The proposed approach first uses a FCM to discover a latent variable that may exist in the dataset and then uses that value namely *FCMout* as an additional input in the classification models both in the training and testing phase to improve a systems accuracy. Moreover, the proposed methodology aims to predict the class of a data object with high accuracy so that this class can be used by a RS to produce recommendations. Finally, it produces a list of recommendations based exclusively only on the class of each object.

The work presented in this part of our research lead to the publication of two conference papers (Christodoulou et al., 2017) and (Christodoulou et al., 2017).

Chapter 6: Session-Based Recommendations on Sequence-Based Data

6.1 Introduction

Recommender Systems (RS) constitute a key part of modern ecommerce websites (Schafer et al., 1999); their aim is to enhance the user experience, by providing personalized product recommendations. Recent study on RS has mainly focused on matrix factorization (MF) methods and neighborhood search-type models. Such approaches work well in practice when a rich user profile can be built from the available data. Unfortunately, though, rich user profiles seldom are readily available to real-world systems.

Session-based recommendation is a characteristic challenge that cannot be properly addressed by conventional methodologies employed in the context of RS. Specifically, under a session-based setup, recommendation is based only on the actions of a user during a specific browsing session (Schafer et al., 1999). Indeed, this type of recommendation generation approach is based on tracking user actions during an active session. Based on the captured and inferred session-based user behavioral patterns, it tries to predict the following user actions during that session, and proactively recommend items/actions to them.

From this description, it becomes apparent that session-based recommendation engines attempt to generate effective recommendations with the availability of user-specific data being extremely limited. Consequently, under this setting, conventional algorithmic approaches towards RS are confronted with hard challenges that stem from the unavailability of rich user profiles (data sparsity) (Koren, 2008). Hence, in order to obtain effective session-based RS, it is imperative that novel methodological approaches be devised. Such methods must be capable of more effectively inferring and leveraging subtle session patterns, with the ultimate goal of enriching the available user profiles so as to properly address the challenges associated with data sparsity.

Indeed, user session data constitute action sequences potentially entailing rich, complex, and subtle temporal dynamics. Thus, enabling effective extraction of these underlying dynamics, and utilizing them in the context of a preference inference mechanism, may result in novel session-based RS with considerably improved recommendation quality performance compared to the alternatives. Markov chain models (Shani et al., 2005) constitute the most typical type of machine learning methods used to achieve this goal. However, recent breakthroughs in the field of Deep Learning (DL) (LeCun et al., 2015) have lately come into close scrutiny, as a potential alternative means of addressing these challenges. Specifically, the introduction of novel treatments of Recurrent Neural Networks (RNNs) (Bengio et al., 2013), with compelling performance in as challenging and diverse tasks as image recognition, natural language understanding, and video captioning, has motivated their application to session-based RS. Contrary to simple deep network formulations that comprise only feed forward connections, RNNs also entail recurrent connections that allow for them to construct an internal representation of their observations history (Rumelhart et al., 1986). This representation, which is encoded in the form of a high-dimensional vector of hidden unit activations, can then be utilized to address challenging learning tasks dealing with sequential data.

In this context, the work recently presented in (Hidasi et al., 2015) constitutes the most characteristic development. The RNNs employed therein are presented with data regarding the items a user selects and clicks on during a given session. On this basis, recommendation relies on the history of previous actions (clicks on items) during that session, and the inferred behavioral patterns. As shown therein, this method yields state-of-the-art session-based recommendation performance in several challenging benchmark problems.

Motivated from these advances, in this thesis we seek derivation of a solid inferential framework that allows for increasing the capability of RNN-driven session-based RS to ameliorate the negative effects of data sparsity. To this end, we draw inspiration from recent RS developments which rely on the utilization of Bayesian inference techniques (e.g., (Chatzis, 2012, Chatzis, 2013, Harvey et al., 2011). Bayesian inference in the context of RS can be performed by considering that the postulated model variables pertaining to the system users and items are stochastic latent variables with some prior distribution imposed over them. This inferential framework allows for the developed recommendation engine to account for the uncertainty in the available (sparse) training data. Thus, it is expected to allow for much improved predictive performance outcomes compared to the alternatives.

Under this rationale, our proposed approach is founded upon the fundamental assumption that the hidden units of the postulated RNNs constitute latent variables of stochastic nature, imposed some appropriate prior distribution. On this basis, we proceed to infer their corresponding posteriors, using the available training data. Specifically, to allow for our model to scale to real-world datasets, comprising millions of examples, we perform inference by resorting to the amortized variational inference (AVI) paradigm (Kingma & Welling, 2014, Kingma et al., 2014). This is an approximate inference approach, which consists in: (i) parameterizing the inferred posterior distributions by means of conventional neural networks (inference networks); and (ii) casting the inference problem as an optimization problem, by making use of ideas from variational calculus.

We evaluate the efficacy of the so-obtained approach, dubbed Recurrent Latent Variable Network for Session-Based Recommendation (ReLaVaS), considering a challenging publicly available benchmark. We compare the obtained predictive performance of ReLaVaS with state-of-the-art techniques; we show that our approach completely outperforms the competition, without presenting any limitations in terms of computational efficiency and scalability.

6.2 Related Work

Recently, several authors have considered introducing elaborate statistical assumptions into MF, that allow for performing full Bayesian inference (e.g., (Chatzis, 2012, Chatzis, 2013, Harvey et al., 2011). Under this approach, it is considered that the user and item variables constitute stochastic latent variables, over which an appropriate prior distribution is imposed, and a corresponding posterior is inferred from the data. Broad empirical evidence has shown that, under such a Bayesian inference-driven setup, real-world RS can yield a noticeable predictive accuracy improvement without compromises in computational scalability. Indeed, this outcome is well-expected from a theoretical point of view; this is due to the fact that a Bayesian inference treatment allows for better accounting for the uncertainty in the (training) data, which is prevalent in RS due to the sparsity of the available ratings matrices.

On the other hand, in the last years the field of machine learning has witnessed a new wave of innovation, due to the Deep Learning (DL) breakthrough (LeCun et al., 2015). Unsurprisingly, the significant advances accomplished in the context of DL have had a noteworthy impact on the ongoing research on RS. Indeed, several researchers working on model-based CF methods have recently proposed novel CF algorithms that employ DL-based models as an alternative to conventional MF-driven approaches.

In this vein, the work of Salakhutdinov et al. (2007) constitutes one of the earliest ones that adopted ideas from the field of DL to affect the CF task. Specifically, they employed Restricted Boltzmann Machines (RBMs) to learn the user and item latent vectors, and showed that their approach outperforms various popular alternatives in the Netflix challenge dataset. More recently, the method in (Wang et al., 2015) presented a hierarchical Bayesian model called collaborative deep learning (CDL) by augmenting the MF algorithm with appropriate side information related to item content. This side information is obtained, in turn, from a DL model; this learns to extract useful, high-level representations from the raw item content, so as to inform the MF algorithm.

Despite this extensive research effort devoted to RS, session based recommendation is a field that remained unappreciated for quite a long time, and has only recently attracted significant attention from the research community. Indeed, most of news and media sites, as well as many e-commerce sites (especially of small retailers), track the users that visit their sites only for short periods of time. Besides, the use of cookies or browser fingerprinting does not allow for obtaining reliable user data over long periods, spanning multiple sessions. Finally, it is very often the case that the behavior of users exhibits session-based traits.

These facts bring to the fore the need of developing effective session-based RS, that can satisfy the following desiderata: (i) each session of the same user must be treated independently of their previous ones; (ii) the used algorithms must be capable of extracting subtle temporal behavioral patterns from the available user proles, e.g. item-to-item similarity, co-occurrence, and transition probabilities; and (iii) this inferential procedure must be effectively carried out over long temporal horizons, as opposed to unrealistic low-order (e.g., one-step) temporal dependence models, that take only the last click or selection of the user into account (and ignore the information of past clicks in the same session).

To address these issues, (Zhang et al., 2014) introduced a novel framework based on traditional RNNs, and evaluated it using the click-through logs of a large scale commercial engine; their results showed significant improvements on the click-prediction accuracy compared to sequence-independent approaches. In a similar vein, Hidasi et al., (2015) presented an RNN-type machine learning model capable of learning subtle temporal patterns in user session data obtained from large ecommerce websites. Specifically, to allow for effectively extracting high-order temporal dynamics, they utilized Gated Recurrent Unit (GRU) networks (Cho et al., 2014). Such networks entail a more elaborate model of an RNN unit, which aims at dealing with the vanishing/exploding gradient problem; this is a problem that plagues training of conventional RNNs, often rendering it completely infeasible (Hochreiter & Schmidhuber, 1997). Their method was shown to outperform state-of-the-art alternatives in two large-scale tasks, including the challenging RecSys Challenge 2015 benchmark (Ben-Shimon et al., 2015). Finally, Tan et al., (2016) proposed two extensions of the breakthrough work of Hidasi et al., (2015), namely: (i) data augmentation via sequence preprocessing; and (ii) a simple model pre-training technique, to account for temporal shifts in the data distribution. As they showed, their proposed extensions yield an improvement over the method in (Hidasi et al., 2015) by more than 10%.

6.3 Proposed Approach

The main contribution of this part of our research consists in devising a machine learning model capable of extracting temporal dynamics from sparse user session data and then utilizing this information to produce accurate recommendations. The ReLaVaS formulates the generation session-based recommendations as a sequence-based prediction problem. Let us denote a user session where x_i is the i^{th} clicked item; then, we may formulate session-based recommendations as the problem of predicting the score vector of the available items, where $y_{i+1,j}$ is the predicted score of the j^{th} item. We are interested in recommending more than one item, therefore, at each time point we select the *top-k* items to recommend back to the user. The goal of this approach is to devise a machine learning model for predicting accurately vector y_{i+1} given the observed subsequences.

6.3.1 Methodological Background

Our model is inspired by the state-of-the-art RNN-based method presented in Hidasi et al., (2015) and relies on an RNN structure that utilizes GRU units. At each time point, i, the postulated network is presented with the current user action (selected item), x_i, and is expected to generate a prediction for the score vector y_{i+1} pertaining to the (i +1)th user selection. The recurrent units' activation vectors of the GRU-based network, $\boldsymbol{h}$, are updated at time i according to equation (43):

$$h_i = (1 - z_i) \cdot h_{i-1} + z_i \cdot \widehat{h_\iota} \tag{43}$$

where **.** donates the elementwise product between two vectors, h_{i-1} is the activation vector of the recurrent units at the previous time point and z_i is the update gate output, which essentially controls when and by how much to update the hidden state of the recurrent units; z_i is presented in equation (44).

$$z_i = \tau(W_z x_i + U_z h_{i-1}) \tag{44}$$

where $\tau()$ is the logistic sigmoid function and the W_z and U_z are trainable network parameters. On the other hand, in equation 43 $\widehat{h}_i$ is the candidate activation vector of the GRU units at time i. This expression is a standard recurrent unit update function with trainable variables W, U and is presented in equation (45).

$$\widehat{h_\iota} = \tanh\,(W x_i + U(r_i \cdot h_{i-1})) \tag{45}$$

Here r_i is the output of the reset gate of the GRU network outlined in equation (46), which is trained to decide when the internal memory of the GRU units must be reset, with the ultimate goal of preventing the gradients of the model objective function from exploding to infinity or vanishing to zero during training.

$$r_i = \tau(W_r x_i + U_r h_{i-1}) \tag{46}$$

W_r and U_r are again trainable network parameters.

6.3.1.1 Model Formulation

The proposed model extends the principles discussed in the previous section; it introduces a novel approach that renders the GRU-based model manageable by a Bayesian inference. We consider the component recurrent unit activations as stochastic latent variables and we start by imposing a prior distribution over them as shown in equation (47):

$$p(h_i) = N(h_i | 0, I) \tag{47}$$

where $N(\xi|\mu,\Sigma)$ is a multivariate Gaussian density with mean μ, covariance matrix Σ and identity matrix I.

Furthermore, we seek to devise an efficient mean of inferring the corresponding posterior distributions, given the available training data. To this end, we drawinspiration by the AVI paradigm (Kingma, & Welling, 2014); we postulate that the sought posteriors $q(h)$ take the form of Gaussians with means and isotropic covariance matrices parameterized via GRU networks as shown in equation (48):

$$q(h_i;\theta) = N(h_i|\mu_\theta(x_i), \sigma_\theta^2(x_i)I) \tag{48}$$

In the above formula the mean vectors $\mu_\theta(x_i)$ and the variance functions $\sigma_\theta^2(x_i)$are outputs of the GRU network with parameters θ resulting in a new expression as seen in equation (49):

$$[\mu_\theta(x_i), log\sigma_\theta^2(x_i)] = (i - z_i)\cdot[\mu_\theta(x_{i-1}), log\sigma_\theta^2(x_{i-1})] + z_i\cdot\widehat{h_\imath} \tag{49}$$

where

$$z_i = \tau(W_z x_i + U_z[\mu_\theta(x_{i-1}), log\sigma_\theta^2(x_{i-1})]) \tag{50}$$

$$\widehat{h_\imath} = \tanh\,(Wx_i + U(r_i\cdot[\mu_\theta(x_{i-1}), log\sigma_\theta^2(x_{i-1})])) \tag{51}$$

$$r_i = \tau(W_r x_i + U_r[\mu_\theta(x_{i-1}), log\sigma_\theta^2(x_{i-1})]) \tag{52}$$

$[\xi, \zeta]$ presents the concatenation of vectors ξ and ζ. The values of the latent variables h_i can be computed by posterior samples from the inferred posterior density.

Let us continue with the output layer of the proposed model. In general, item ranking can be pointwise, pairwise or listwise. Pointwise ranking estimates the score of items independently of each other; then, the goal of model training is to ensure that relevant items receive a high score. Pairwise ranking compares the score of pairs of a positive and a negative item; then, model training aims at enforcing the score of the positive item to be higher than that of the negative one, for all the available pairs. Such a construction allows for one to limit score computation for the purposes of model training to a select subset of the available items. On the downside, such a formulation may undermine the eventually obtained accuracy of the recommendation algorithm. On the other hand, pointwise approaches require score computation for the whole set of available items. This is certainly more computationally demanding than pairwise approaches. However, this extra computational complexity does not necessarily translate into reduced scalability to real-world systems. This is especially the case with DL algorithms, which can be easily parallelized at a large scale by using cheap GPU hardware. Finally, listwise ranking uses the scores of all items and compares them to the perfect ordering. This entails item sorting, which can be computationally prohibitive in cases of large-scale systems. Motivated from this discussion, in this work we resort to the most straightforward conditional likelihood selection for our model, namely a simple Multinoulli distribution; that is

$$p(y_{i+1,j} = 1|h_i)\alpha\tau(w_y^j \cdot h_i) \tag{53}$$

w_y are trainable parameters of the output layer of the model.

Concerning the negative log-likehood function selection of the proposed model, L_s, we utilize Bayesian Personalized Ranking (BPR) pairwise ranking loss presented in Rendle et al., (2009). This technique compares the score of the positive item with several sampled items and then utilizes their average as the loss. The loss is defined at a given point in a session as shown in equation (54).

$$L_s = -\frac{1}{N_s} E\left[\sum_{j=1}^{N_a} \log\left(\tau(\hat{y}_{s,i} - \hat{y}_{s,j})\right)\right] \tag{54}$$

where $\hat{y}_{s,k}$ is the predictive score on item *k* at the given point of the sth session, *i* is the next item in the session while *j* are the negative samples at a given point of a session. Finally, N_s is the number of negative samples.

6.3.1.2 Training Algorithm

Let us consider a training dataset *D,* which comprises a number of click sequences (sessions), pertaining to a multitude of users. The variational inference (Jordan et al., 1998) for the proposed model consists in maximization of a lower-bound to the log-marginal likelihood of the model (evidence lower-bound, ELBO). Based on the previous model formularization, the ELBO expression of the ReLaVaS is shown in equation (55):

$$\log \mathrm{p}(D) \geq \sum_i \{-KL[q(h_i;\theta)||p(h_i)] - E[L_S]\} \tag{55}$$

where $KL[q||p]$ is the KL divergence between the distribution $q(\cdot)$ and the distribution $p(\cdot)$ and is presented in equation (56).

$$KL[q(h_i;\theta)||p(h_i)] = -\frac{1}{2}\sum_{d=1}^{D}[\mu_\theta(x_i)^{\,2}]_{\,d} + \frac{D}{2}[1 + log\sigma_\theta(x_i)^{\,2} - \sigma_\theta(x_i)^{\,2}] \quad (56)$$

Unfortunately, the posterior expectation $E[L_S]$ cannot be computed analytically. This is due to the non-conjugate formulation of the proposed model, which stems from its nonlinear assumptions. As a result, training the entailed parameter sets θ is not possible. AVI resolves these issues by means of a smart re-parameterization of the Monte Carlo (MC) (Salakhutdinov & Mnih, 2011) samples of a Gaussian posterior density (Kingma & Welling, 2014) as shown in equation (57).

$$h^\gamma = \mu_\theta(\cdot) + \sigma_\theta(\cdot)\varepsilon^\gamma \quad (57)$$

where ε^γ is the white random noise with unitary variance. The MC samples resulted from the posterior density can be now expressed as differentiable functions of the parameters sets θ and some random noise variance ε. As a result, the problematic posterior expectation $E[L_S]$ reduces to a much more attractive posterior, a low variance (random noise) variable. Then, by taking the gradient of the ELBO in the context of any stochastic optimization algorithm, one can yield low variance estimators for the parameter sets. To this end Kingma et al., (2014) suggest the use of Adagrad as the stochastic gradient algorithm with adaptive step-size (Duchi et al., 2011), so we also follow this advice and select Adagrad as the stochastic optimizer for training the proposed model.

6.3.1.3 Prediction Generator

Having trained a ReLaVaS model, given some dataset D, recommendation generation in the context of a user session can be performed by computing the predicted ratings $\boldsymbol{y}_{i+1}$, and selecting the top-k of them to recommend to the user. To effect this procedure, we sample the latent variables $\boldsymbol{h}_i$ from the corresponding variational posterior distributions. Indeed, to allow for obtaining reliable estimators, we draw a set comprising Γ samples from the posteriors; eventually, this yields a set of scoring function samples. Then, recommendation is performed on the basis of the mean of these samples; that is, we employ a standard MC-type rationale.

6.3.2 Experimental Setup

To provide strong empirical evidence of the merits of our approach, in the following we extensively evaluate it in challenging experimental scenarios. To obtain some comparative results, apart from our method we also examine the performance of existing GRU networks, which constitute the recently proposed state-of-the-art alternative that is closest to our approach. In addition, we also assess the performance of a standard baseline method, different for each experimental scenario. Our source codes have been developed in Python, and made use of the Theano library (Bastien et al., 2012). We run our experiments on an Intel Xeon 2.5GHz Quad-Core server with an NVIDIA Tesla K40 GPU accelerator.

6.3.3 ReLaVaS Model Configuration

In the following, we experiment with a diverse set of selections for the size of the latent variable space (number of recurrent latent variables). In all cases, parameter initialization for our model is performed by resorting to the Glorot Normal initialization scheme (Glorot & Bengio, 2010); dropout with a rate equal to 0.5 is employed for regularization purposes. To perform model training, Adagrad is carried out by utilizing session-parallel mini-batches, following the suggestions in Hidasi et al., (2015).

Let us consider we adopt a mini-batch size equal to β. Then, session-parallel mini-batches can be obtained by using the first event of the first sessions to form the input data of the first mini-batch (the desired output is the second events of our active sessions); we use the second events to form the second mini-batch, and so forth. When a session ends, we put the next available session in its place. In the occasion of such a switch taking place, we reset the appropriate hidden state of the model, since we assume that the training sessions constitute independent and identically distributed (sequential) data. To facilitate convergence, Nesterov momentum (Qian, 1999) is additionally applied during training. In all cases, Adagrad's global stepsize is chosen from the set {0:005; 0:01; 0:05; 0:1} , while momentum strength is chosen from the set {0; 0:1; 0:2; 0:3; 0:4}, both on the basis of the obtained performance on the training set in the first few training algorithm iterations. Finally, concerning the negative loglikelihood function selection of our model as already mentioned, we use a BPR loss function.

6.3.4 Performance metrics

To quantitatively assess the performance of our approach, we employ two commonly used evaluation metrics, namely Recall@20 and Mean Reciprocal Rank (MRR)@20. The former metric expresses the frequency at which the desired (groundtruth) item in the test data makes it to the 20 highest ranked items suggested by the evaluated approach. Hence, this metric allows for modeling and assessing certain practical scenarios where there is no highlighting of recommendations; what matters is the desired item being included in a short list of recommendations, rather than the absolute order that these items are presented to the user. On the other hand, MRR@20 describes the average predicted score of the desired items in the test data, with the score values set to zero if the desired item does not make it to the top-20 list of ranked items. Thus, MRR@20 models scenarios where absolute item ordering does matter; for instance, it allows for better algorithm evaluation in cases where the lower ranked items are visible only after scrolling.

6.3.5 RecSys Challenge 2015 dataset

To evaluate our method in session-based recommendation, we exploit the benchmark dataset released in the context of the RecSys Challenge 2015 (Ben-Shimon et al., 2015); this comprises click-stream data pertaining to user sessions with an e-commerce website. Unfortunately, the test set of the aforementioned benchmark does not provide groundtruth information that can be used for recommendation quality evaluation. To resolve this issue, we adopt the solution suggested in Hidasi et al., (2015); we split the originally available training data into one set comprising 7,966,257 sessions (with a total of 31,637,239 click actions), and another one comprising the remainder 5,324 sessions (with a total of 71, 222 click actions); we use the former for model training and the latter for evaluation purposes. Both sets entail a total of 37,483 items that a user may select to click on. Thus, we are dealing with a very sparse dataset, where the need of inferring more subtle and informative temporal patterns comes to the fore with increased complexity.

6.3.5.1 Empirical Performance

We commence the presentation of our experimental results by reporting on the best-performing configuration of our model (i.e., selection of the number of latent variables that maximizes empirical performance on the test set); our findings are summarized in

Table 31. In the same Table, we also illustrate how these empirical findings compare to the conventional GRU-driven approach reported in Hidasi et al., (2015), two variants of it reported in Tan et al., (2016), namely the M2 and M4 methods, as well as a baseline method in recommendation systems, namely BPR-MF (Rendle et al., 2009). We report two different performance results for the GRU-driven approach, which correspond to two different loss functions considered in Hidasi et al., (2015), namely BPR and TOP1; the former selection yields better Recall@20 outcomes for that method, while the latter yields a better MRR@20 value. As we observe, our approach outperforms all previously reported state-of-the-art results in terms of both the Recall@20 metric and the MRR@20 metric. We obtained these results with the mini-batch size equal to 50, the dropout value set to 0.5, the learning rate set to 0.1 and momentum equal to 0.3.

Table 31: Best performance results of the evaluated methods

Method	Hidden Units	Recall@20	MRR@20
BPR-MF	-	0.2574	0.0618
GRU w/ BPR Loss	1000	0.6322	0.2467
GRU w/ TOP1 Loss	1000	0.6206	0.2693
M2	100	0.7129	0.3091
M4	1000	0.6676	0.2847
ReLaVaS	**750**	**0.7971**	**0.7845**

6.3.5.2 Varying the size of the latent variable space

It is well-understood that the number of latent units bears significant impact on the obtained empirical performance. To allow for examining the extent of this effect, in Figure 18 we show how the considered performance metrics vary when adjusting the number of latent units of a trained ReLaVaS model. These results have been obtained with the training algorithm hyperparameters remaining the same. As it can be seen from Figure 19, ReLaVaS accuracy, as measured via both the considered metrics, is low for small model sizes, grows substantially for larger models, but quickly reaches a plateau, after which it starts to deteriorate.

Figure 19: ReLaVaS w/ BPR Loss performance fluctuation with the number of latent variables

6.3.5.3 Considering alternative loss functions

Further, it is interesting to examine how the performance of our model compares to the competition in case we adopt a different type of negative log-likelihood function, L_s. To this end, we consider the TOP1 loss function that was introduced in Hidasi et al., (2015), as well as a standard cross-entropy loss function also mentioned in the aforementioned work. The obtained results (for best model configuration) are provided in Table 32. As we observe, appropriate selection of the employed negative log-likelihood function is a crucial factor that determines the success of our approach in modeling the considered dataset. To provide some further insights, in Figures 20 and 21 we illustrate the corresponding results regarding performance fluctuation with the size of the latent variable space. We observe that model size continues to have a significant effect on ReLaVaS predictive performance when using these alternative negative log-likelihood functions.

Table 32: ReLaVaS model performance for different selections of the negative log-likehood function

Loss Function	BPR	TOP1	Cross-Entropy
# Latent Units	750	1000	1500
Step Size	0.1	0.1	0.05
Momentum	0.3	0	0
Recall@20	0.7971	0.6250	0.6507
MRR@20	0.7845	0.2727	0.3527

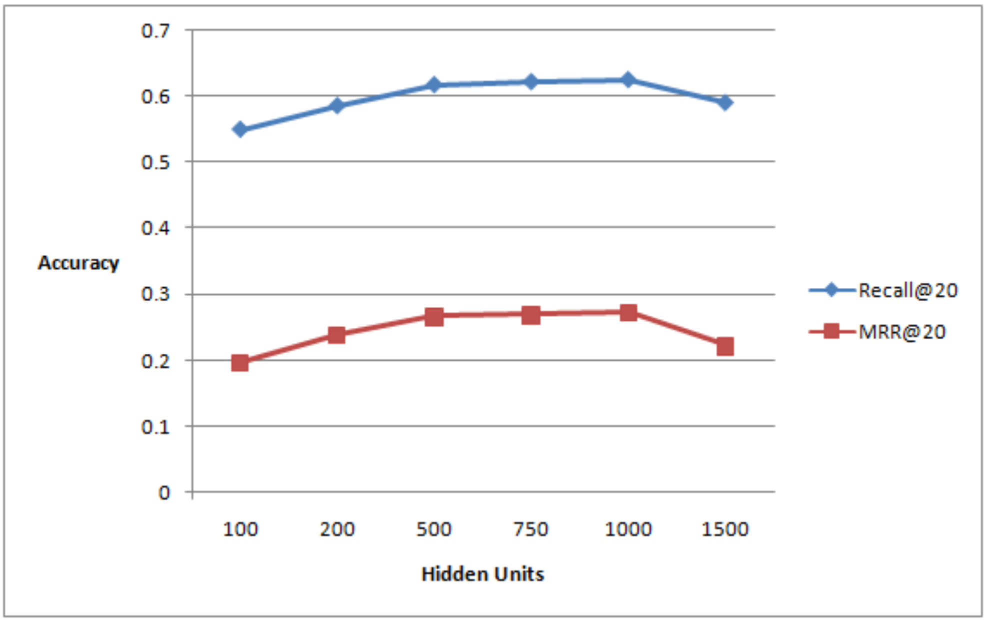

Figure 20: ReLaVaS performance fluctuation with the number of latent variables, when employing the TOP1 loss function

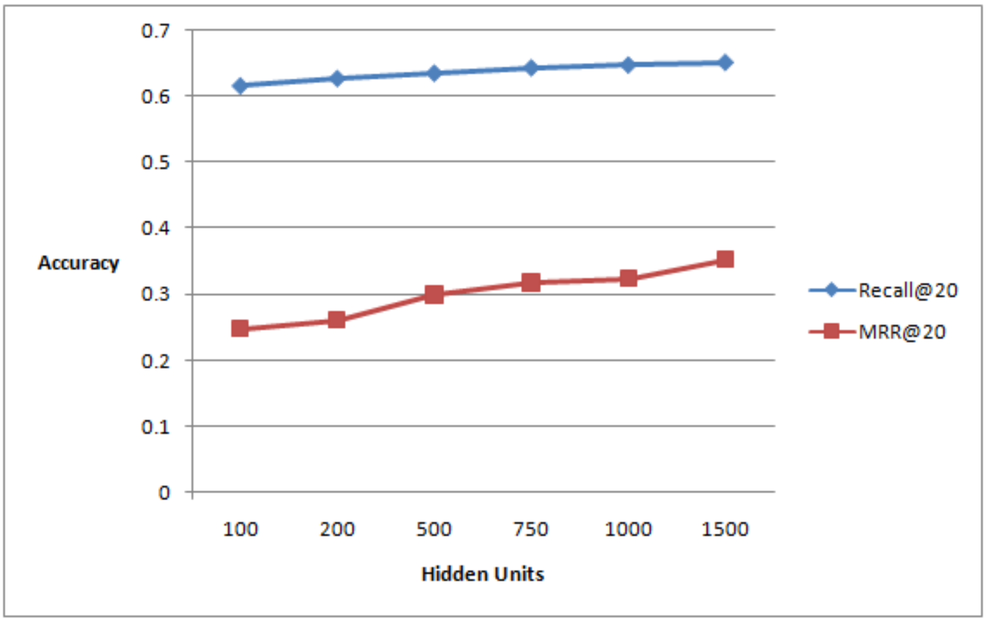

Figure 21: ReLaVaS performance fluctuation with the number of latent variables, when employing the Cross-Entropy loss function

6.3.5.4 *Computational Complexity*

Apart from predictive accuracy, another aspect of machine learning models that is of utmost importance when dealing with real-world applications concerns computational efficiency. This aspect entails examination of both model scalability to large datasets, as well as of the imposed computational overheads when it comes to prediction generation. To investigate these aspects of the proposed approach, in Table 33 we perform a comparison of wall-clock times between our method and the current state-of-the-art. The measurements reported therein pertain to network sizes yielding the best empirical accuracy in each case. Further, in Figures 22-27, we show how computational complexity of the proposed model varies with model size (for the examined loss functions). From this exhibition, it becomes apparent that not only our approach yields competitive accuracy, but it does so without substantially increasing the computational costs. Specifically, notice that the slight difference in computational costs between the original formulation of ReLaVaS (i.e., using the BPR loss function) and the competition is merely due to the smaller size of the trained network. Note also that all the compared approaches allow for real-time prediction generation. Thus, we can soundly argue that our method constitutes an attractive solution for building real-world predictive systems for data with temporal dynamics.

Table 33: Comparison of computational times (in seconds) for various selections of loss functions.

Method	Network Size	Training time	Prediction time per click event (average)
GRU w/ BPR Loss	1000	48692.48	0.683
GRU w/ TOP1 Loss	1000	44716.60	0.627
ReLaVaS w/ TOP1 Loss	1000	42357.84	0.595
ReLaVaS w/ Cross-Entropy Loss	1500	60109.86	0.844
ReLaVaS w/ BPR Loss	750	43009.75	0.604

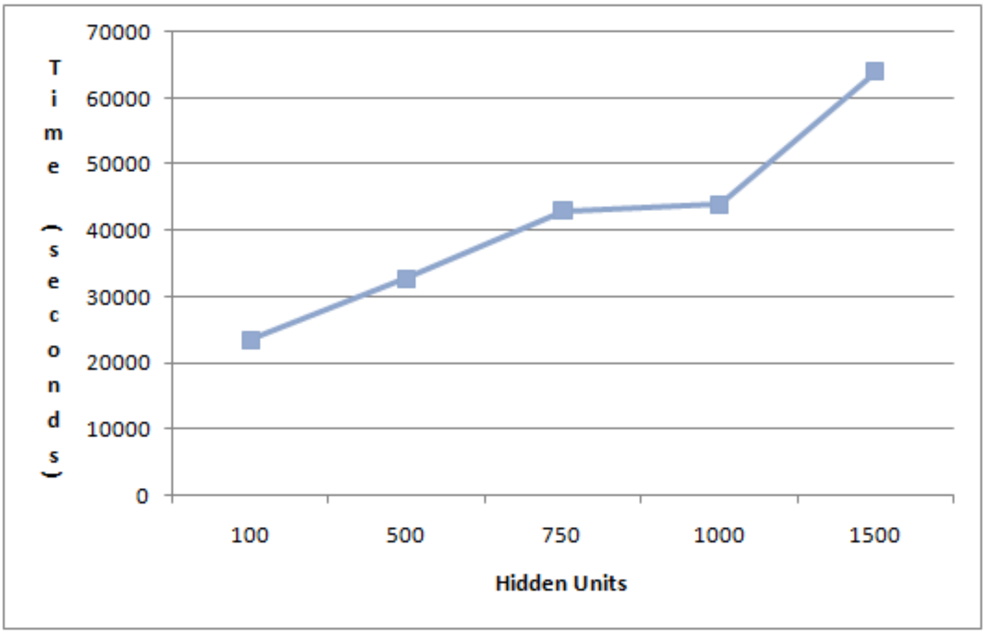

Figure 22: ReLaVaS w/ BPR Loss total training time fluctuation with network size (in seconds)

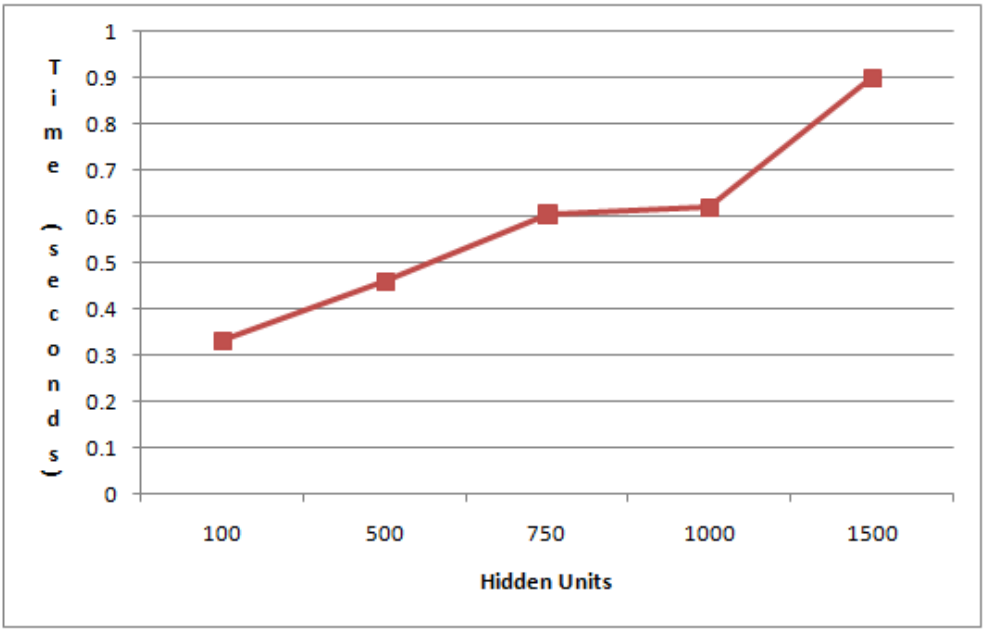

Figure 23: ReLaVaS w/ BPR Loss average prediction time fluctuation with network size (in seconds)

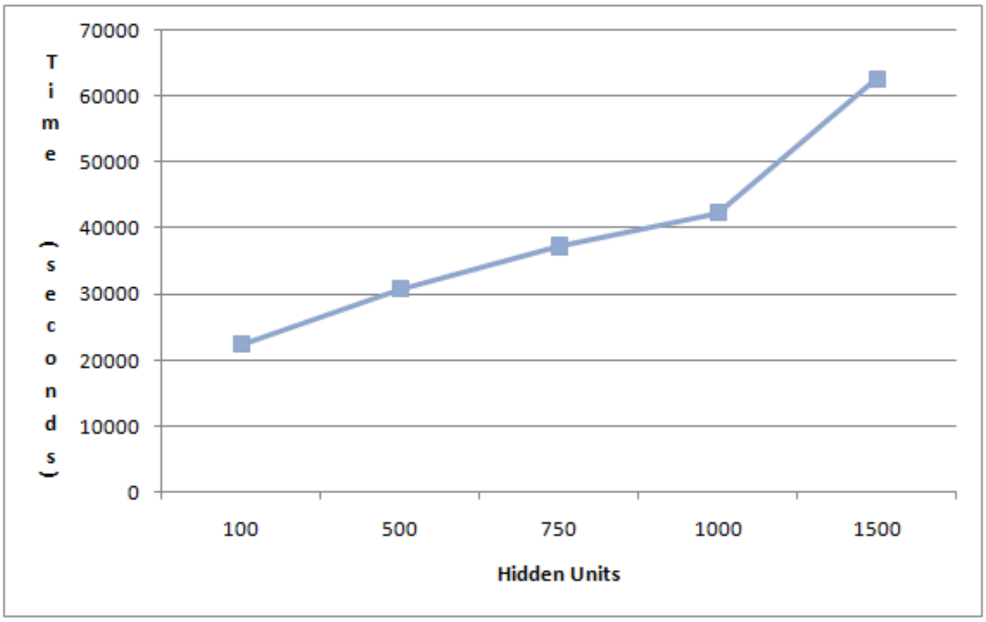

Figure 24: ReLaVaS w/ TOP1 Loss total training time fluctuation with network size (in seconds)

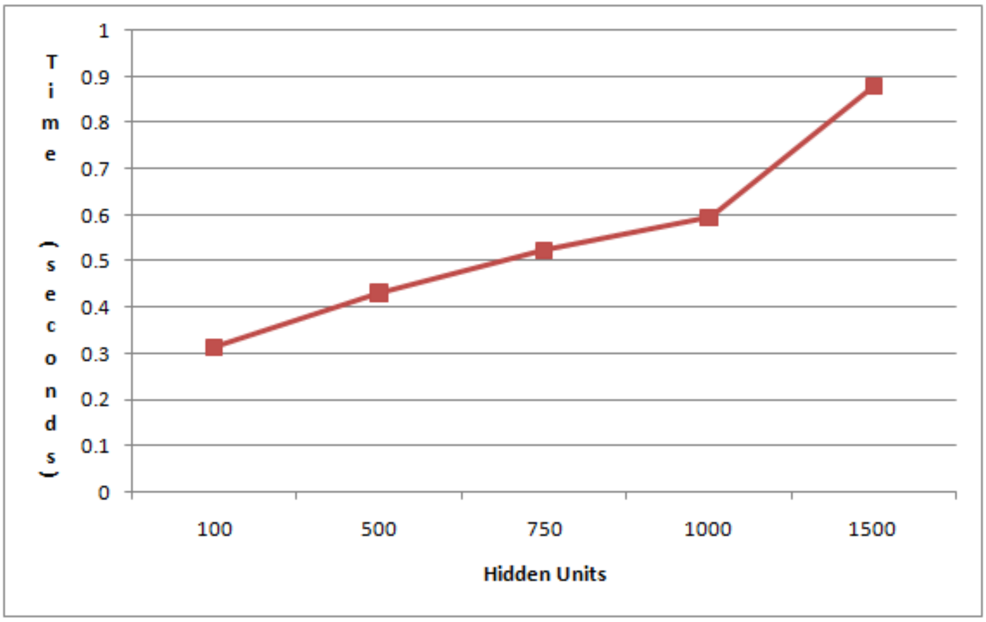

Figure 25: ReLaVaS w/ TOP1 Loss average prediction time fluctuation with network size (in seconds)

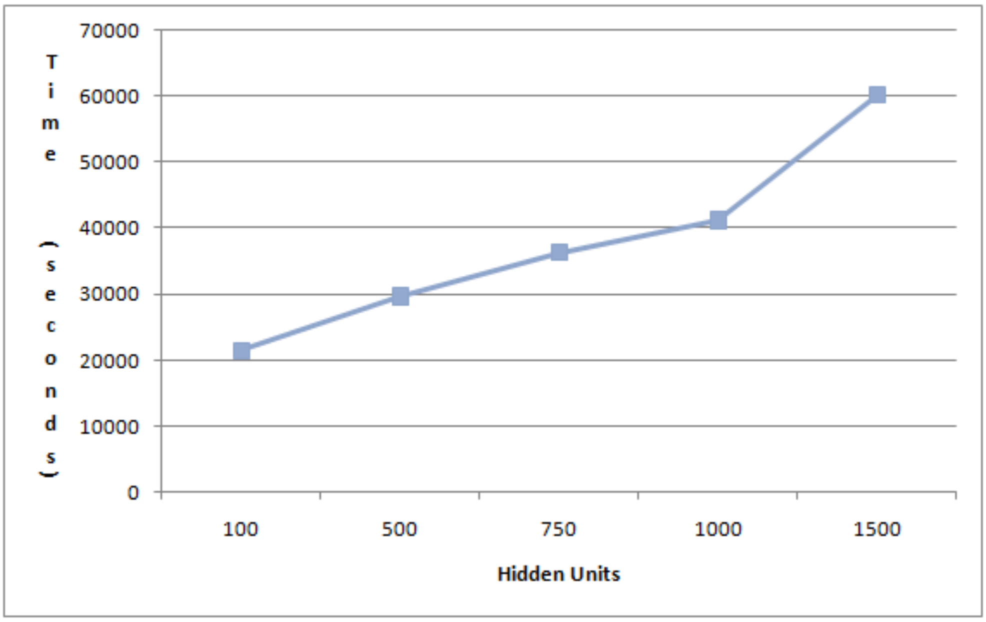

Figure 26: ReLaVaS w/ Cross-Entropy Loss total training time fluctuation with network size (in seconds)

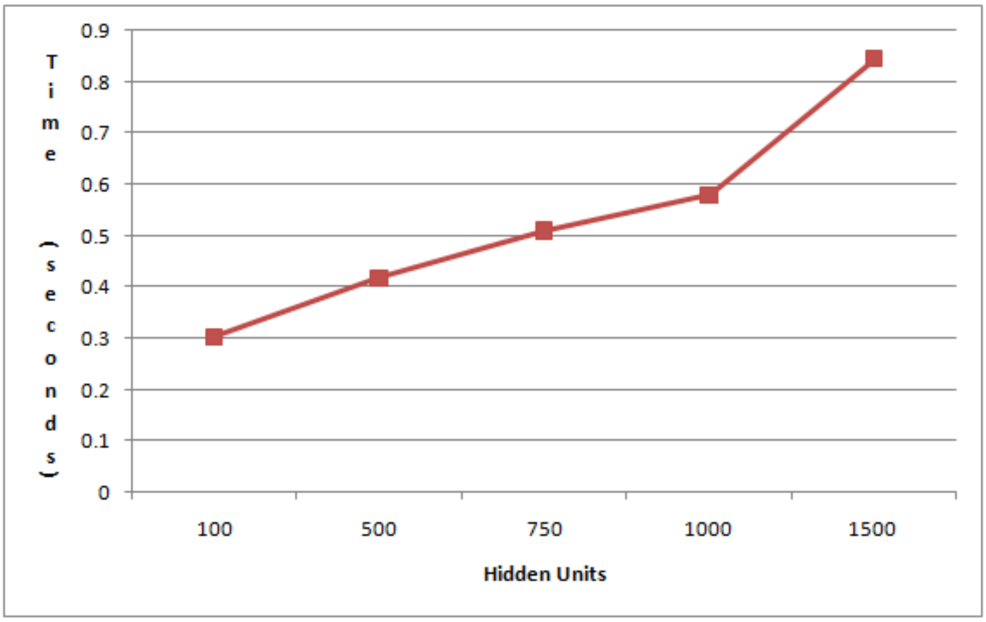

Figure 27: ReLaVaS w/ Cross-Entropy Loss average prediction time fluctuation with network size (in seconds)

6.3.6 Taxi Service Trajectory Prediction

To further illustrate the novelty of the proposed methodology in the context of a different application area, in this section we take advantage of the Taxi Service Trajectory dataset that was released for the ECML PKDD 2015 Prediction Challenge (Discovery Challenge, 2015). This dataset includes the trajectories performed by all the 442 taxis running in the city of Porto, Portugal.

The goal of this challenge is to predict, at any given time point, the trajectory a taxi service will follow until it reaches its destination, given the trajectory it has followed so far. To perform our experimental evaluations under different settings, we split the available dataset into two different sets, a small one and a large one. This allows for us to evaluate our approach under scenarios with different qualitative characteristics (limited training data as well as larger training datasets).

The small dataset comprises the available samples from the 1st of July until the 20th of August, 2014. We filter these data to only keep location items that appear at least 5 times in the dataset, and in at least two taxi service examples. Then, we split the resulting dataset into a training set, which covers the period until the 25th of July, and a test set which comprises the remainder of the available data. This results in a training set that comprises 100,453 location visits, 429 taxi service examples and 201 location items, and a test set that consists of 4,068 location visits, 392 taxi service examples and 49 location items.

Similar is the case with the compilation of the large dataset, except that we use the available samples from the 1st of July until the 12th of November, 2014. The training set covers the period until the 12th of October. In constructing our test set, we only retain sessions that comprise more than 2,200 location visits, to allow for evaluating our model in problems dealing with long sequences. This way, our resulting training set comprises 619,945 location visits, 436 taxi service examples and 503,355 location items, while the test set includes 122,947 location visits, 50 taxi service examples and 103,522 location items.

6.3.6.1 Small Dataset: Varying the size of the latent variable space

In Tables 34-35, we show the performance statistics of standard GRU networks and the proposed ReLaVaS approach, for the various considered model size alternatives. As we observe, ReLaVaS outperforms the state-of-the-art GRU approach with respect to both the considered performance metrics.

Table 34: Small Dataset: GRU model performance

Hidden Units	Recall@20	MRR@20
100	0.81746	0.30292
1000	0.8079	0.23486
1500	0.85006	0.24082
2000	0.88418	0.30298

Table 35: Small Dataset: ReLaVaS model performance

Hidden Units	Recall@20	MRR@20
100	0.82280	0.30402
1000	0.94604	0.30644
1500	0.94908	0.30732
2000	0.94112	0.39324

The best obtained empirical results of each method are summarized in Table 36. Therein, we report two different selections of model size for the ReLaVaS model. This is due to the fact that Recall@20 yields its best value when using 1500 units, while MRR@20 achieves its best when using 2000 units.

Table 36: Small Dataset: Best results of the evaluated approaches

Method	Model Size	Recall@20	MRR@20
ItemKNN	-	0.0606	0.018
GRU	2000	0.88418	0.30298
ReLaVaS	1500 / 2000	0.94908	0.39324

We observe that the proposed ReLaVaS model achieves an accuracy improvement of 7.3% on the Recall@20 metric, and 29.7% on the MRR@20 metric.

Turning to the baseline ItemKNN method, it suffices to simply inspect Figure 28 to observe that algorithms incapable of modeling subtle temporal dynamics are completely inappropriate for dealing with the considered application.

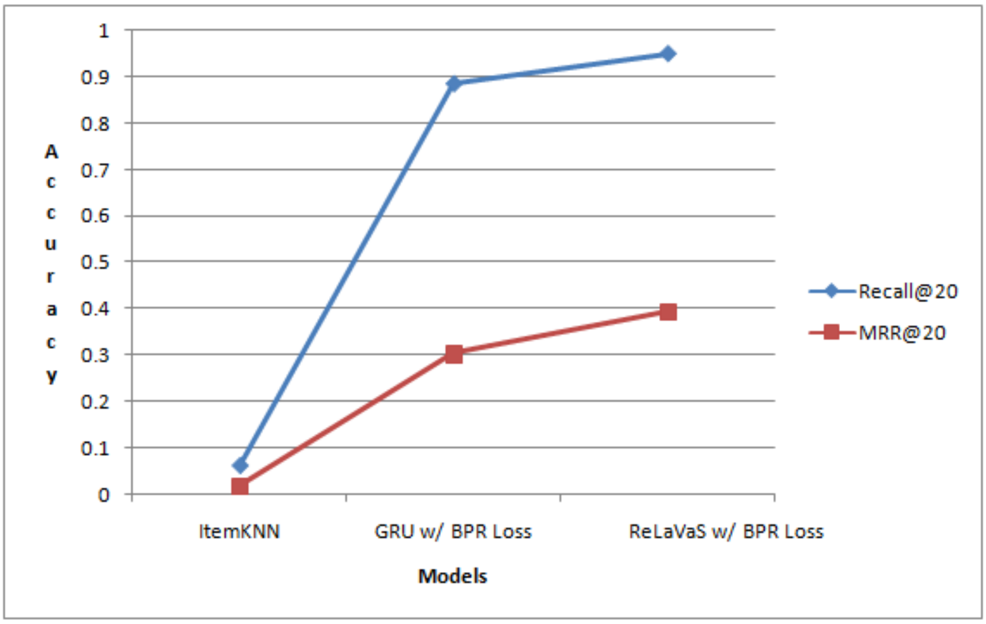

Figure 28: Small Dataset: Performance of various competitors against ReLaVaS

6.3.6.2 Small Dataset: Computational Complexity

In order to investigate how our method compares to the competition in terms of computational costs, we perform a comparison of wall-clock times between our approach and standard GRU networks, for the best performing model configurations mentioned in the previous section. The recorded measurements that concern total training time and average prediction time are reported in Table 37.

For completeness sake, we also plot the fluctuation of ReLaVaS computational costs with model size in Figures 29 (total training time) and 30 (average prediction time). We observe that the proposed method not only produces better accuracy results, but it achieves this without imposing additional computational costs.

Table 37: Small Dataset: Comparisons of computational times (in seconds)

Method	Network Size	Training time	Prediction time per click event (average)
GRU	2000	251.45	0.06197
ReLaVaS	1500	165.25	0.04073
ReLaVaS	2000	253.35	0.06241

Figure 29: Small Dataset: ReLaVaS total training time fluctuation with network size (seconds)

Figure 30: Small Dataset: ReLaVaS average prediction time fluctuation with network size (seconds)

6.3.6.3 Large Dataset: Varying the size of the latent variable space

Similar to the small dataset, in Tables 38-39 we outline the performance outcomes of standard GRU networks and the proposed ReLaVaS approach, in the case of the large dataset; the best results of each method (obtained for optimal model configuration) are summarized in Table 40.

Note that, in this scenario, we have not considered model sizes greater than 150 units; this is due to the associated requirements in RAM memory, which rendered it infeasible for us to train larger GRU networks and ReLaVaS models.

Table 38: Large Dataset: GRU model performance

Hidden Units	Recall@20	MRR@20
50	0.79266	0.54298
75	0.84494	0.68676
100	0.84204	0.67700
150	0.79404	0.57082

Table 39: Large Dataset: ReLaVaS model performance

Hidden Units	Recall@20	MRR@20
50	0.80746	0.56616
75	0.86656	0.73398
100	0.82058	0.64252
150	0.79518	0.60268

We again observe that the proposed ReLaVaS model outperforms GRU networks on both the considered performance metrics. It achieves an increase of 2.5% on the Recall@20 metric, and of 6.8% on the MRR@20 metric.

Table 40: Large Dataset: Best results of the evaluated approaches

Method	Model Size	Recall@20	MRR@20
ItemKNN	-	0.0428	0.0112
GRU	75	0.84494	0.68676
ReLaVaS	75	0.86656	0.73398

Similar to the previous experiment, Figure 31 shows that the ItemKNN method is completely incapable of producing meaningful performance outcomes.

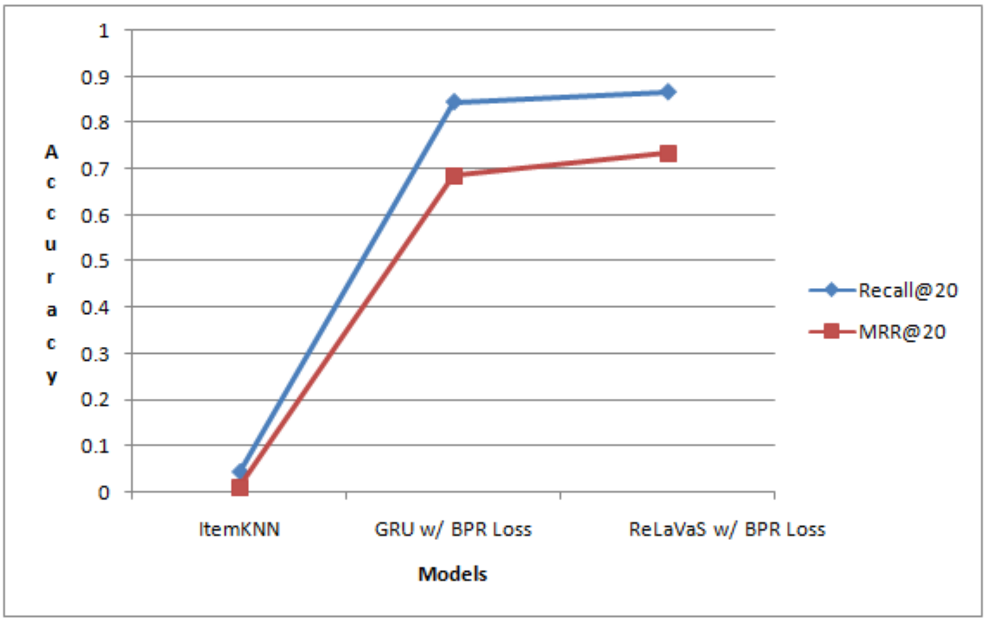

Figure 31: Large Dataset: Performance of various competitors against ReLaVaS

6.3.6.4 Large Dataset: Computational Complexity

Finally, similar to the small dataset, we perform a comparison of wall-clock times between our approach and standard GRU networks, for the best performing model configurations reported in Table 40. The recorded measurements that concern total training time and average prediction time are reported in Table 41.

The fluctuation of ReLaVaS total training time is shown in Figure 32, while the fluctuation of ReLaVaS average prediction time is presented in Figure 33. As we observe, ReLaVaS computational costs are comparable to the state-of-the-art, which our method greatly outperforms in terms of predictive accuracy.

Table 41: Large Dataset: Comparisons of computational times (in seconds)

Method	Network Size	Training time	Prediction time per click event (average)
GRU	75	176.23	0.005353
ReLaVaS	75	174.59	0.008103

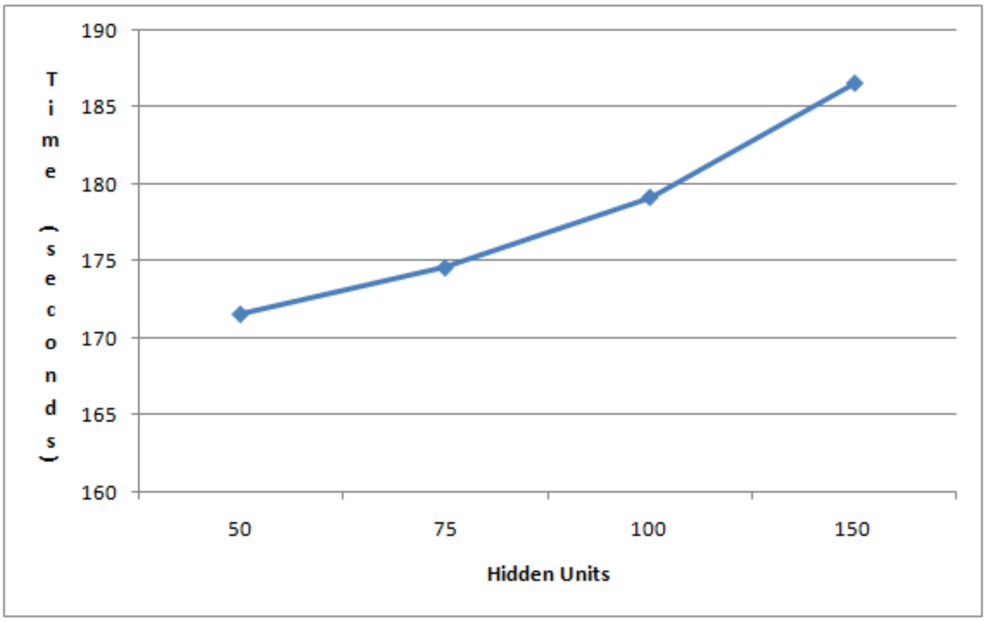

Figure 32: Large Dataset: ReLaVaS total training time fluctuation with network size (seconds)

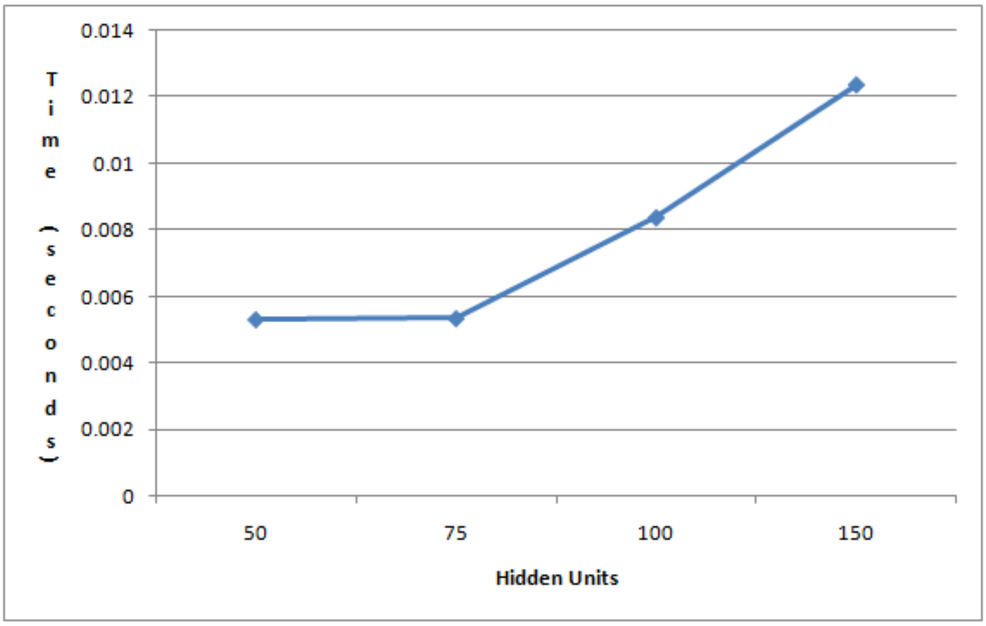

Figure 33: Large Dataset: ReLaVaS average prediction time fluctuation with network size (seconds)

6.3.7 Overview

This chapter introduced a Recurrent Latent Variable Network for Session-Based Recommendation (ReLaVaS) that deals with sequence-based data. The proposed work is one of the first studies on the field of Deep Learning RS and aims to tackle the data sparsity problem in session-based recommendations where the classic models were not able to deal with in the past. The proposed model extends the principles of the the state-of-the-art RNN-based method that relies on an RNN structure and utilizes GRU units by introducing a Bayesian inference that is able to render the GRU-based approach. Our methodology was compared with the current state-of-the-art techniques on two different datasets from various domains (e-commerce and GPS data) outperforming completely the competition, without presenting any limitations in terms of computational efficiency and scalability.

The work described in the final part of this thesis lead to the publication of two conference papers (Christodoulou et al., 2017), (Christodoulou et al., 2017) and to the submission of a journal paper (Christodoulou et al., 2017) that is under review.

Chapter 7: CONCLUSIONS

This thesis introduces new concepts and methodologies for RS aiming to enhance the user experience and at the same time to improve the system's accuracy by dealing with the challenges of RS. Table 42 presents the contributions of this work as well as the approaches used and the outcomes of each methodology.

Table 42: PhD contributions list

Contribution	Approach	Outcome
Use of an entropy-based approach and clustering algorithms	Entropy-based approach, Hard K-Modes, Fuzzy K-Modes	Overcome the cold-start and data sparsity limitation
Adoption of a rule-based system to discover sub-datasets of items	Mapping method, rule-based techniques based on specific characteristics	Face scalability issues
Introduction of a Bayesian Inference model	Bayesian Inference model	Predict whether an item is likely to be recommended or not
Use of machine learning techniques in combination with Fuzzy Cognitive Maps	Fuzzy Cognitive Maps Support Vector Machines, Linear Discrimination, weighted k-NN, Classification Trees, Hybrid models	Improve the system's accuracy
Implementation of deep learning models on session-based/sequence-based data	Variational Recurrent Neural Network	Deal with data sparsity

The first part of our research concentrates more on the development of new Multi-criteria RS aiming to improve the accuracy and performance of RS by utilizing various techniques such as an entropy-based approach, clustering methods and Bayesian inference models. The proposed methodology practically eliminates data sparsity and scalability issues due to the nature of its recommendation engine.

More specifically, the proposed movie RS does not require ratings of items prior to providing recommendations; it only needs to learn the interests of a user during the learning session. Thus, as long as this session is relatively short, cold start will not be an issue. In addition, the information captured during the learning sessions was combined with the search input that is provided by the user in real time, with his/her preferences being dynamically adjusted in a continuous manner. Therefore, this process also handles sparsity as the volume of the available data is reduced (actually split) during the formation of the clusters, which takes place when the system is first launched or when newer data samples may become available. Scalability is always an issue with every RS and the proposed system is not an exception. Preliminary experimentation, with varying size of the dataset (i.e. working with increasing portions of the available dataset) suggested that clustering is the most time consuming process, followed by the combination of the three categories of information (movie categories, stars, production companies) in the construction of the recommendation list, and that computational time rises proportionally to the size of the clusters formed. Nevertheless, safer conclusions may be drawn, and in general inference will be made possible, only after a more thorough investigation of the effect of size and/or type of information recorded in the datasets is executed. A series of synthetic experiments were executed to validate the proposed RS system on the MovieLens1M dataset. More specifically, five different users with various interests on movies performed ten different searches and the recommendations yielded by the Hard and Fuzzy K-Modes algorithms were assessed in terms of accuracy using the well-known RMSE metric. Overall, the recommendations yielded by the proposed RS were accurate, with the results revealing small superiority of the Fuzzy over the Hard implementation. The proposed RS was also compared with two implementations of the well-known KNN algorithm, a simple one and one enhanced with the dynamic information of a user's recent searches. The comparison suggested that both clustering approaches produced more accurate recommendations than the two KNN variations and also proved that the use of dynamic information in the RS engine is a key parameter for improving its performance.

In the same context we introduced a real-time targeted RS deployed in a supermarket setting where users are constantly changing their preferences or shopping habits, and products potentially change their characteristics or lose their popularity. We argued that there is a need for a system that captures the dynamic environment of a supermarket aiming to recommend products that are on offer. Throughout this part we discussed how the cold-start problem, the data sparsity and other scalability issues often met in RS are minimized by utilizing an Entropy-based Hard k-modes clustering methodology. Moreover, using a Bayesian inference approach we demonstrated how the system can suggest personalized recommendations to users considering different pieces of evidence.

To explore whether our methodology meets our expectations, we deployed the proposed methodology in two departments of a supermarket where customers installed the prototype mobile application on their smart devices. Feedback was obtained on the recommendations made for each customer and used for evaluating the system's accuracy. Our findings outperformed the traditional CF approaches (item-based and user-based).
The second part of this thesis dealt with the classification prediction problem in order to solve some of the challenges of RS. This part provided a clear methodology able to enhance the capabilities of classic classification models combined with FCMs to develop hybrid models that can lead to better accuracy results. This work introduced a methodology that aims to improve the accuracy of classification models when used for prediction purposes. The proposed approach coupled Fuzzy Cognitive Maps (FCM) with classic classification models constructing hybrid models that can lead to better performance and prediction abilities. The optimal performance of the hybrid models in terms of accuracy depends on two things: Firstly, the dataset used must be clear and understandable consisting of variables that have numerical continuous values and linear/monotonic relations so that the identification of the correlations that exist between different attributes is more profound. Second, there is enough knowledge available (i.e. through experts or literature) to enable the construction of the correct map that describes well the problem under study. Moreover, a recommendation engine demonstrated how classification prediction models can produce exclusive recommendations based only on the assigned class of an object tackling at the same time the cold-start and scalability issues.

In the final part of this thesis, we tackled the problem of session-based recommendation by designing a novel deep learning RNN model. Specifically, our work was motivated from the sequential nature of the addressed predictive setup, and the associated sparsity of the available data. Our expectation was that, by better addressing these issues, one may be able to obtain a noticeable improvement in the quality of the generated recommendations. We introduced a way of improving the modeling capacity of state-of-the-art GRU networks by adopting concepts from the field of Bayesian statistics, namely variational inference. The proposed approach, dubbed ReLaVaS, constitutes a hierarchical latent variable model, where the inferred posterior distributions are parameterized via GRU networks. Such a Bayesian inferential setup: (i) retains the prowess of existing GRU-based networks in terms of extracting and analyzing salient temporal patterns in the available sequential data; and (ii) allows for accounting for the uncertainty in the available (sparse) data when performing prediction generation; enabling this capability is well-known to yield a noticeable performance improvement in real-world data modeling scenarios. As we showed, our approach is capable of outperforming existing state-of-the-art alternatives in terms of two popular performance metrics. We also illustrated that our proposed approach achieves this accuracy improvement without undermining computational efficiency, both in training time and in prediction generation time.

7.1 Future Work

Future work will concentrate more to improve the effort done in this thesis as well as to expand the work on new domains of application utilizing new trends and technologies on Recommender Systems.

Although our preliminary experimentation suggested the validity of the Multi-criteria methodologies, additional experimentation needs to be undertaken to validate the accuracy and performance of the proposed system in supermarkets of a larger scale; in terms of productsand customers. In addition, the parameter that controls the number of recommended products needs to be automatically tuned to further increase the system's accuracy. Furthermore, an investigation needs to be undertaken on how the number of products influences the number of clusters discovered. Finally, experiments will be carried out to study how efficiently the system propagates notifications through iBeacons and how does group recommendations affect buyers in purchasing products.

Regarding the work done on second part of this thesis the work will focus on (i) further investigating the performance of those models utilizing other types of datasets which contain labelled variables and/or nonlinear relations, (ii) improving the hybrid models' performance by using learning techniques for automating and optimizing the construction of the FCM part and the definition of causalities, both in terms of strength and direction and (iii) implementing a RS for producing real-time recommendations to newly registered users.

One research direction that we have not considered in the Session-based recommendations concerns the possibility of stacking multiple network layers, one on top of the other, to create a more potent sequential data modeling pipeline. In such a formulation, the input of the bottom GRU layer is the observed data sequence, on the other hand, each one of the subsequent GRU layers is presented with the sequence of activation vectors, of the layer that immediately precedes it in the hierarchy; the model output layer is driven from the recurrent unit activations vector of the top most GRU layer. Stacking multiple layers of GRU networks allows for performing inference and analysis of temporal patterns in multiple time-scales. Moreover, we want to investigate the outcomes of the proposed methodology when increasing the number of the sampled items in the output as well as when performing data augmentation or pre-training on the available datasets.

In addition, we should also consider the possibility of implementing the proposed approaches on a blockchain framework. A blockchain is a decentralized, public ledger that contains a list of records, a blockchain can continuously grow overtime forming a group of blocks linked together. Blockchain will aim to increase the security and anonymity of recommendations increasing people' privacy and trust for the RS. This technology will also allow us to use a distributed environment to compute the recommendations overcoming scalability issues rewarding at the same time users that help in the maintenance of the network.

Finally, we will seek ways to apply the methodologies presented in this work in new domains of application such as in the area of smart manufacturing to optimize a concept/object production line. A brief idea is to utilize our approaches on a car manufacturing production line. We can use various sensors to monitor and record any available data from various sources at each step of the production line with the aim to produce recommendations that can reduce manufacturing costs and time.

REFERENCES

Aart F. de Vos. (2008). A primer in Bayesian Inference.

Adomavicius, G., Kwon, Y. (2007). New recommendation techniques for multicriteria rating systems. IEEE Intelligent Systems 22 (3) , pp. 48-55.

Adomavicius, G., Kwon, Y. (2012). Improving aggregate recommendation diversity using ranking-based techniques. IEEE Transactions on Knowledge and Data Engineering 24 (5) , art. no. 5680904 , pp. 896-911.

Adomavicius, G., Mobasher, B., Ricci, F., Tuzhilin, A. (2011). Context-aware recommender systems. AI Magazine 32 (3) , pp. 67-80.

Adomavicius, G., Mobasher, B., Ricci, F., Tuzhilin, A. (2011). Context-aware recommender systems. AI Magazine 32 (3) , pp. 67-80.

Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. ACM Transactions on Information Systems 23 (1) , pp. 103-145.

Adomavicius, G., Tuzhilin, A. (2005). A survey of the state-of-the-art and possible extensions (2005). Towards the next generation of recommender systems. IEEE Transactions on Knowledge and Data Engineering 17 (6) , pp. 734-749.

Adomavicius, G., Tuzhilin, A. (2005). A survey of the state-of-the-art and possible extensions (2005). Towards the next generation of recommender systems. IEEE Transactions on Knowledge and Data Engineering 17 (6) , pp. 734-749.

Adomavicius, Gediminas and YoungOk Kwon, (2011). New Recommendation Techniques for Multi-Criteria Rating Systems In Recommender systems handbook, Springer US, pp. 769-803

Agarwal, D., Chen, B.-C. (2010). fLDA: Matrix factorization through latent dirichlet allocation. WSDM 2010 - Proceedings of the 3rd ACM International Conference on Web Search and Data Mining, pp. 91-100.

Anand, D., Bharadwaj, K.K. (2011). Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities. Expert Systems with Applications 38 (5) , pp. 5101-5109.

Andreou, A.S., Mateou, N.H. & Zombanakis, G.A., (2005). Soft computing for crisis management and political decision making: the use of genetically evolved fuzzy cognitive maps. Soft Computing, 9(3), pp.194–210.

Baltrunas, L., Makcinskas, T., Ricci, F. (2010). Group recommendations with rank aggregation and collaborative filtering. RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems , pp. 119-126.

Barragáns-Martínez, A.B., Costa-Montenegro, E., Burguillo, J.C., Rey-López, M., Mikic-Fonte, F.A., Peleteiro, A. (2010). A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. Information Sciences 180 (22) , pp. 4290-4311.

Benjamin Heitmann (2010). Using Linked Data to Build Open, Collaborative Recommender Systems. Linked Data Meets Artificial Intelligence Conference

Bobadilla, J., Hernando, A., Ortega, F., Bernal, J. (2011). A framework for collaborative filtering recommender systems. Expert Systems with Applications 38 (12) , pp. 14609-14623.

Bobadilla, J., Ortega, F., Hernando, A., Alcalá, J. (2011). Improving collaborative filtering recommender system results and performance using genetic algorithms. Knowledge-Based Systems 24 (8), pp. 1310-1316.

Bobadilla, J., Serradilla, F., Bernal, J. (2010). A new collaborative filtering metric that improves the behavior of recommender systems. Knowledge-Based Systems 23 (6) , pp. 520-528.

Bollen, D., Knijnenburg, B.P., Willemsen, M.C., Graus, M. (2010). Understanding choice overload in recommender systems. RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems , pp. 63-70.

Bu, J., Tan, S., Chen, C., Wang, C., Wu, H., Zhang, L., He, X. (2010). Music recommendation by unified hypergraph: Combining social media information and music content. MM'10 - Proceedings of the ACM Multimedia 2010 International Conference , pp. 391-400.

Bueno, S. & Salmeron, J.L., (2009). Benchmarking main activation functions in fuzzy cognitive maps. Expert Systems with Applications, 36(3 PART 1), pp.5221–5229.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User Modelling and User-Adapted Interaction 12 (4) , pp. 331-370.

Cacheda, F., Carneiro, V., Fernández, D., Formoso, V. (2011). Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. ACM Transactions on the Web 5 (1), art. no. 2.

Candanedo, L.M. & Feldheim, V., (2016). Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. Energy and Buildings, 112, pp.28–39.

Canny, J. (2002). Collaborative filtering with privacy via factor analysis. SIGIR Forum (ACM Special Interest Group on Information Retrieval) , pp. 238-245.

Cantador, I., Bellogín, A., Vallet, D. (2010). Content-based recommendation in social tagging systems. RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems, pp. 237-240.

Cantador, I., Konstas, I., Jose, J.M. (2011). Categorising social tags to improve folksonomy-based recommendations. Journal of Web Semantics 9 (1) , pp. 1-15.

Chatzis, P. S., Christodoulou, P., & Andreou, A. S. (2017). Recurrent Latent Variable Networks for Session-Based Recommendation. 2nd Deep learning workshop – RecSys 2017.

Chen, J., Nairn, R., Nelson, L., Bernstein, M., Chi, E. (2010). Short and tweet: Experiments on recommending content from information streams. Conference on Human Factors in Computing Systems - Proceedings 2 , pp. 1185-1194.

Chiu, P.-H., Kao, G.Y.-M., Lo, C.-C. (2010). Personalized blog content recommender system for mobile phone users. International Journal of Human Computer Studies 68 (8), pp. 496-507.

Cho, Y.H., Kim, J.K. (2004). Application of Web usage mining and product taxonomy to collaborative recommendations in e-commerce. Expert Systems with Applications 26 (2) , pp. 233-246.

Cho, Y.H., Kim, J.K., Kim, S.H. (2002). A personalized recommender system based on web usage mining and decision tree induction. Expert Systems with Applications 23 (3) , pp. 329-342.

Christodoulou, P., Chatzis, P. S., & Andreou, A. S. (2017). A Recurrent Latent Variable Model for Supervised Modeling of High-Dimensional Sequential Data. IEEE Transactions on Neural Networks and Learning Systems.

Christodoulou, P., Chatzis, P. S., & Andreou, A. S. (2017). A Variational Recurrent Neural Network for Session-Based Recommendations using Bayesian Personalized Ranking. 26th International Conference on Information Systems Development.

Christodoulou, P., Christodoulou, K., & Andreou, A. S. (2017). A Real-Time Targeted Recommender System for Supermarkets. 19th International Conference on Enterprise Information Systems.

Christodoulou, P., Christoforou, A., & Andreou, A. S. (2017). A Hybrid Prediction Model Integrating Fuzzy Cognitive Maps With Support Vector Machines. 19th International Conference on Enterprise Information Systems.

Christodoulou, P., Christoforou, A., & Andreou, A. S. (2017). Improving the Performance of Classification Models with Fuzzy Cognitive Maps. 2017 IEEE Conference on Fuzzy Systems.

Christodoulou, P., Lestas, M., & Andreou, A. S. (2013). A Dynamic Web Recommender System Using Hard and Fuzzy K-Modes Clustering. In IFIP International Conference on Artificial Intelligence Applications and Innovations (pp. 40-51). Springer, Berlin, Heidelberg.

Christodoulou, P., Lestas, M., & Andreou, A. S. (2014). Applying hard and fuzzy K-modes clustering for dynamic Web recommendations. Engineering Intelligent Systems, 2014, Volume 22, Issue 3-4, Pages 177-190.

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A.C. and Bengio, Y.(2015). A recurrent latent variable model for sequential data. In Advances in neural information processing systems, pp. 2980-2988

Constantinos, S., Andreas, A. (2007). A Hybrid Software Component Clustering and Retrieval Scheme, Using an Entropy-based Fuzzy k-Modes Algorithm. In Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence, vol 1, pp. 202-209

Cortes, C. & Vapnik, V., (1995). Support-Vector Networks. Machine Learning, 20(3), pp.273–297

Cremonesi, P., Koren, Y., Turrin, R. (2010). Performance of recommender algorithms on top-N recommendation tasks. RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems, pp. 39-46.

D. DeCoste. (2006). Collaborative prediction using ensembles of maximum margin matrix factorizations. In Proc. ICML'06. Pittsburgh, Pennsylvania, USA, 249–256.

D. Kingma and M. Welling. (2014). Auto-Encoding Variational Bayes. In Proc.ICLR'14.

D. P. Kingma, D. J. Rezende, S. Mohamed, and M.Welling. (2014). Semi-Supervised Learning with Deep Generative Models. In Proc. NIPS'14.

D.E. Rumelhart, G.E. Hinton, and R.J. Williams. (1986). Learning internal representations by error propagation. In Parallel Dist. Proc. MIT Press, 318–362.

David Ben-Shimon, Alexander Tsikinovsky, Michael Friedmann, Bracha Shapira, Lior Rokach, and Johannes Hoerle. (2015). RecSys Challenge 2015 and the YOOCHOOSE Dataset. In Proceedings of the 9th ACM Conference on Recommender Systems (RecSys ’15). ACM, New York, NY, USA, 357–358.

De Campos, L.M., Fernández-Luna, J.M., Huete, J.F., Rueda-Morales, M.A. (2010). Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. International Journal of Approximate Reasoning 51 (7) , pp. 785-799.

Discovery Challenge: On Learning from Taxi GPS Traces: ECML–PKDD. (2015). [Online]. Available: http://www.geolink.pt/ecmlpkdd2015-challenge

Duan, L., Street, W.N., Xu, E. (2011). Healthcare information systems: Data mining methods in the creation of a clinical recommender system. Enterprise Information Systems 5 (2) , pp. 169-181.

Duchi, J., Hazan, E. and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 12(Jul), pp.2121-2159

F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio. (2012). Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.

Gantner, Z., Drumond, L., Freudenthaler, C., Rendle, S., Schmidt-Thieme, L. (2010). Learning attribute-to-feature mappings for cold-start recommendations. Proceedings - IEEE International Conference on Data Mining, ICDM , art. no. 5693971 , pp. 176-185.

Gao, M., Wu, Z., Jiang, F. (2011). Userrank for item-based collaborative filtering recommendation. Information Processing Letters 111 (9) , pp. 440-446.

Garcia Esparza, Sandra, Michael P. O' Mahony, and Barry Smyth. (2011). A multi-criteria evaluation of a user generated content based recommender system. In Proceedings of the 3rd ACM RecSys’ 10 Workshop on Recommender Systems and the Social Web

García-Crespo, A., López-Cuadrado, J.L., Colomo-Palacios, R., González-Carrasco, I., Ruiz-Mezcua, B. (2011). Sem-Fit: A semantic based expert system to provide recommendations in the tourism domain. Expert Systems with Applications 38 (10) , pp. 13310-13319.

Garcia, I., Sebastia, L., Onaindia, E. (2011). On the design of individual and group recommender systems for tourism. Expert Systems with Applications 38 (6) , pp. 7683-7692.

Gavalas, D., Kenteris, M. (2011). A web-based pervasive recommendation system for mobile tourist guides. Personal and Ubiquitous Computing 15 (7) , pp. 759-770.

Ge, M., Delgado-Battenfeld, C., Jannach, D. (2010). Beyond accuracy: Evaluating recommender systems by coverage and serendipity. RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems , pp. 257-260.

George Karypis (2001). Evaluation of Item-Based Top-N Recommendation Algorithms. In Proceedings of the 10th international conference on Information and Knowledge Management, pp. 247-254, New York, USA

Georgios Alexandridis, Georgios Siolas and Andreas Stafylopatis, (2013). Improving Social Recommendations by applying a Personalized Item Clustering policy. In Proceedings of ACM RecSys Workshop on Recommender Systems and the Social Web

Goldberg, K., Roeder, T., Gupta, D., Perkins, C. (2001). Eigentaste: A Constant Time Collaborative Filtering Algorithm. Information Retrieval 4 (2) , pp. 133-151.

Gong, S. (2010). A collaborative filtering recommendation algorithm based on user clustering and item clustering. Journal of Software 5 (7) , pp. 745-752.

Gou, J. et al., (2012). A new distance-weighted k-nearest neighbor classifier. J. Inf. Comput. Sci.

Guan, Z., Wang, C., Bu, J., Chen, C., Yang, K., Cai, D., He, X. (2010). Document recommendation in social tagging services. Proceedings of the 19th International Conference on World Wide Web, WWW '10 , pp. 391-400.

Guido Jan de Nooij (2008). Recommender Systems: An Overview", MSc Thesis, University of Amsterdam, November

Guy Shani, David Heckerman, and Ronen I. Brafman. (2005). An MDP-Based Recommender System. J. Machine Learning Research 6 (2005), 1265–1295.

Guy, I., Zwerdling, N., Ronen, I., Carmel, D., Uziel, E. (2010). Social media recommendation based on people and tags. SIGIR 2010 Proceedings - 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 194-201.

Hannon, J., Bennett, M., Smyth, B. (2010). Recommending Twitter users to follow using content and collaborative filtering approaches. RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems, pp. 199-206.

He, Q., Pei, J., Kifer, D., Mitra, P., Giles, L. (2010). Context-aware citation recommendation. Proceedings of the 19th International Conference on World Wide Web, WWW '10 , pp. 421-430.

Herlocker, J.L., Konstan, J.A., Riedl, J. (2000). Explaining collaborative filtering recommendations. Proceedings of the ACM Conference on Computer Supported Cooperative Work, pp. 241-250.

Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T. (2004). Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems 22 (1) , pp. 5-53.

Hidasi, B., Karatzoglou, A., Baltrunas, L., and Tikk, D. (2015). Session-based recommendations with recurrent neural networks. CoRR, abs/1511.06939.

Hofmann, T. (2004). Latent semantic models for collaborative filtering. ACM Transactions on Information Systems 22 (1) , pp. 89-115.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. Data mining and knowledge discovery, 2(3), 283-304.

Huang, Z., Chen, H., Zeng, D. (2004). Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. ACM Transactions on Information Systems 22 (1) , pp. 116-142.

Huang, Zhexue, and Michael K. Ng. (1999). A fuzzy k-modes algorithm for clustering categorical data." *Fuzzy Systems, IEEE Transactions on* 7.4: 446-452.

Hwang, C.-S. (2010). Genetic algorithms for feature weighting in multi-criteria recommender systems. Journal of Convergence Information Technology 5 (8) , pp. 13.

Hwang, C.-S. (2010). Genetic algorithms for feature weighting in multi-criteria recommender systems. Journal of Convergence Information Technology 5 (8) , pp. 13.

Hwang, Cheinshung, and Tungsheng Chang. (2012). Genetic k-means collaborative filtering for multi-criteria recommendation. Journal of Computational Information Systems 8, no. 1, pp. 293-303

Iakovidis, D.K. & Papageorgiou, E., (2011). Intuitionistic Fuzzy Cognitive Maps for Medical Decision Making. IEEE Transactions on Information Technology in Biomedicine, 15(1), pp.100–107.

J.W. Smith, J.E. Everhart, W. C. Dickson, W.C. Knowler, and R.S. Johannes. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In Proceedings of the Annual Symposium on Computer Application in Medical Care, pp. 261.

Jahrer, M., Töscher, A., Legenstein, R. (2010). Combining predictions for accurate recommender systems. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining , pp. 693-701.

Jason J. Jung, Xuan Hay Pham. (2011). Attribute selection-based recommendation framework for short-head user group: An empirical study by MovieLens and IMDB. In Proceedings of the Third International conference on Computational collective intelligence: Technologies and Applications, pp. 592-601, Berlin, Germany

Jeong, B., Lee, J., Cho, H. (2010). Improving memory-based collaborative filtering via similarity updating and prediction modulation. Information Sciences 180 (5) , pp. 602-612.

Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., and Saul, L.K. (1998) An introduction to variational methods for graphical models. In Learning in Graphical Models, M.I. Jordan (Ed.). Kluwer, Dordrecht, pp. 105–162

Jung, J.J. (2012). Attribute selection-based recommendation framework for short-head user group: An empirical study by MovieLens and IMDB. Expert Systems with Applications 39 (4) , pp. 4049-4054.

Karatzoglou, A., Amatriain, X., Baltrunas, L., Oliver, N. (2010). Multiverse Recommendation: N-dimensional Tensor Factorization for context-aware Collaborative Filtering. RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems , pp. 79-86.

Karypis, G. (2001). Evaluation of item-based top-N recommendation algorithms. International Conference on Information and Knowledge Management, Proceedings , pp. 247-254.

Kazienko, P., Musiał, K., Kajdanowicz, T. (2011). Multidimensional social network in the social recommender system. IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans 41 (4) , art. no. 5766051 , pp. 746-759.

Kim, H.-N., Alkhaldi, A., El Saddik, A., Jo, G.-S. (2011). Collaborative user modeling with user-generated tags for social recommender systems. Expert Systems with Applications 38 (7) , pp. 8488-8496.

Kim, H.-N., Ji, A.-T., Ha, I., Jo, G.-S. (2010). Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation. Electronic Commerce Research and Applications 9 (1) , pp. 73-83.

Kingma, D., and Welling, M. (2014). Auto-Encoding Variational Bayes. In Proc. ICLR'14.
Klitos Christodoulou, Alvaro A. A. Fernandes, Norman W. Paton (2015). Combining Syntactic and Semantic Evidence for Improving Matching over Linked Data Sources. WISE (1) 200-215

Konstan, J.A., Riedl, J. (2012). Recommender systems: From algorithms to user experience. User Modelling and User-Adapted Interaction 22 (1-2) , pp. 101-123.

Konstan, J.A., Riedl, J. (2012). Recommender systems: From algorithms to user experience. User Modelling and User-Adapted Interaction 22 (1-2) , pp. 101-123.

Koren, Y. (2008). Factorization meets the neighborhood: A multifaceted collaborative filtering model. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining , pp. 426-434.

Koren, Y. (2010). Collaborative filtering with temporal dynamics. Communications of the ACM 53 (4) , pp. 89-97.

Koren, Y. (2010). Factor in the neighbors: Scalable and accurate collaborative filtering. ACM Transactions on Knowledge Discovery from Data 4 (1), art. no. 1.

Kosko, B., (1986). Fuzzy cognitive maps. International Journal of Man-Machine Studies, 24(1), pp.65–75.

Kosko, B., (1993). Fuzzy thinking: The new science of fuzzy logic, Hyperion Books.

Kosko, B., (2010). Fuzzy Cognitive Maps: Advances in Theory, Methodologies, Tools and Applications 1st ed. M. Glykas, ed., Springer-Verlag Berlin Heidelberg.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. (2014). On the properties of neural machine translation: Encoder-decoder approaches. In Proc. Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8). ACL, 103–111.

Lacic, E., Kowald, D., Traub, M., Luzhnica, G., Simon, J., and Lex, E. (2015). Tackling cold-start users in recommender systems with indoor positioning systems. In Poster Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16, 2015.

Lam, S.K., Riedl, J. (2004). Shilling recommender systems for fun and profit. Thirteenth International World Wide Web Conference Proceedings, WWW2004 , pp. 393-402.

Lathia, N., Hailes, S., Capra, L., Amatriain, X. (2010). Temporal diversity in recommender systems. SIGIR 2010 Proceedings - 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval , pp. 210-217.

Lawrence, R.D., Almasi, G.S., Koteyar, V., Viveros, M.S., Duri, S.S. (2001). Personalization of supermarket product recommendations. Data Mining and Knowledge Discovery 5 (1-2) , pp. 11-32.

Lee, S.K., Cho, Y.H., Kim, S.H. (2010). Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations. Information Sciences 180 (11) , pp. 2142-2155.

Li, L., Chu, W., Langford, J., Schapire, R.E. (2010). A contextual-bandit approach to personalized news article recommendation. Proceedings of the 19th International Conference on World Wide Web, WWW '10, pp. 661-670.

Li, Q., Wang, J., Chen, Y.P., Lin, Z. (2010). User comments for news recommendation in forum-based social media. Information Sciences 180 (24) , pp. 4929-4939.

Lin, W., Alvarez, S.A., Ruiz, C. (2002). Efficient adaptive-support association rule mining for recommender systems. Data Mining and Knowledge Discovery 6 (1) , pp. 83-105.

Liu, D.-R., Shih, Y.-Y. (2005). Integrating AHP and data mining for product recommendation based on customer lifetime value. Information and Management 42 (3) , pp. 387-400.

Liu, D.-R., Tsai, P.-Y., Chiu, P.-H. (2011). Personalized recommendation of popular blog articles for mobile applications. Information Sciences 181 (9) , pp. 1552-1572.

Liu, J.-G., Zhou, T., Che, H.-A., Wang, B.-H., Zhang, Y.-C. (2010). Effects of high-order correlations on personalized recommendations for bipartite networks. Physica A: Statistical Mechanics and its Applications 389 (4) , pp. 881-886.

Liu, J., Dolan, P., Pedersen, E.R. (2010). Personalized news recommendation based on click behavior. International Conference on Intelligent User Interfaces, Proceedings IUI , pp. 31-40.

Loh, W.-Y., (2011). Classification and regression trees. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1(1), pp.14–23.

Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In Recommender systems handbook (pp. 73-105). Springer US.

Lü, L., Medo, M., Yeung, C.H., Zhang, Y.-C., Zhang, Z.-K., Zhou, T. (2012). Recommender systems. Physics Reports 519 (1) , pp. 1-49.

Lü, L., Medo, M., Yeung, C.H., Zhang, Y.-C., Zhang, Z.-K., Zhou, T. (2012). Recommender systems. Physics Reports 519 (1) , pp. 1-49.

Luo, Ya, and Li Zhao, (2012). Study on Multi-criteria Personalized Recommendation Algorithm. In Proceedings of the 2012 International Conference on Computer Application and System Modeling. Atlantis Press.

Luo, Ya, and Li Zhao, (2012). Study on Multi-criteria Personalized Recommendation Algorithm. In Proceedings of the 2012 International Conference on Computer Application and System Modeling. Atlantis Press.

M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. (1998). An introduction to variational methods for graphical models," in Learning in Graphical Model., M. Jordan, Ed. Dordrecht: Kluwer, pp. 105–162.

Ma, H., King, I., Lyu, M.R. (2011). Learning to recommend with explicit and implicit social relations. ACM Transactions on Intelligent Systems and Technology 2 (3) , art. no. 29.

Ma, H., Zhou, T.C., Lyu, M.R., King, I. (2011). Improving recommender systems by incorporating social contextual information. ACM Transactions on Information Systems 29 (2) , art. no. 9

Manouselis, N., Vuorikari, R., Van Assche, F. (2010). Collaborative recommendation of e-learning resources: An experimental investigation. Journal of Computer Assisted Learning 26 (4) , pp. 227-242.

Manouselis, Nikos, and Constantina Costopoulou. (2007). Analysis and classification of multi-criteria recommender systems. World Wide Web 10, no. 4, pp. 415-441

Massa, P., Avesani, P. (2007). Trust-aware recommender systems. RecSys'07: Proceedings of the 2007 ACM Conference on Recommender Systems , pp. 17-24.

Masthoff, J. (2004). Group modeling: Selecting a sequence of television items to suit a group of viewers. User Modelling and User-Adapted Interaction 14 (1) , pp. 37-85.

McLachlan, G., (2004). Discriminant analysis and statistical pattern recognition (Vol. 544). John Wiley & Sons.

McNee, S.M., Riedl, J., Konstan, J.A. (2006). Being accurate is not enough: How accuracy metrics have hurt recommender systems. Conference on Human Factors in Computing Systems - Proceedings, pp. 1097-1101.

Melville, P., Mooney, R.J., Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. Proceedings of the National Conference on Artificial Intelligence, pp. 187-192.

Michael Ekstrand, John Riedl. (2012). When recommender fail: predicting recommending failure for algorithm selection and combination. In Proceedings of the 6th ACM conference on recommender systems, pp. 233-236

Middleton, S.E., Shadbolt, N.R., De Roure, D.C. (2004). Ontological user profiling in recommender systems. ACM Transactions on Information Systems 22 (1) , pp. 54-88.

Mileta, N. (2015). Recommender systems - a smarter shopping solution using iBeacons

Milicevic, A.K., Nanopoulos, A., Ivanovic, M. (2010). Social tagging in recommender systems: A survey of the state-of-the-art and possible extensions. Artificial Intelligence Review 33 (3) , pp. 187-209.

Miller, B.N., Albert, I., Lam, S.K., Konstan, J.A., Riedl, J. (2003). MovieLens unplugged: Experiences with an occasionally connected recommender system. International Conference on Intelligent User Interfaces, Proceedings IUI , pp. 263-266.

Miller, B.N., Konstan, J.A., Riedl, J. (2004). PocketLens: Toward a personal recommender system. ACM Transactions on Information Systems 22 (3) , pp. 437-476.

Miranda Zhang, Rajiv Ranjan, Surya Nepal, Michael Menzel, Armin Haller (2012). A Declarative Recommender System for Cloud Infrastructure Services Selection. GECON, 102-113

Mobasher, B., Cooley, R., Srivastava, J. (2000). Automatic Personalization Based on Web Usage Mining. Communications of the ACM 43 (8) , pp. 142-151.

Mobasher, B., Dai, H., Luo, T., Nakagawa, M. (2002). Discovery and evaluation of aggregate usage profiles for Web personalization. Data Mining and Knowledge Discovery 6 (1) , pp. 61-82.

Mohandes, M.A. et al., (2004). Support vector machines for wind speed prediction. Renewable Energy, 29(6), pp.939–947.

Mooney, Raymond J., Roy, Loriene. (2000). Content-based book recommending using learning for text categorization. Proceedings of the ACM International Conference on Digital Libraries, pp. 195-204.

Morgan Harvey, Mark J. Carman, Ian Ruthven, and Fabio Crestani. (2011). Bayesian Latent Variable Models for Collaborative Item Rating Prediction. In Proc. CIKM'11. 699–708.

MovieLens datasets: https://grouplens.org/datasets/movielens/

N. Qian. (1999). On the momentum term in gradient descent learning algorithms,"Neural Networks, vol. 12, no. 1, pp. 145–151.

Nikoletic, M. (2013). Recommender systems-a smarter shopping solution. Bachelor's thesis.
Ning, X., Desrosiers, C., & Karypis, G. (2015). A comprehensive survey of neighborhood-based recommendation methods. In Recommender systems handbook (pp. 37-76). Springer US.

Ning, X., Desrosiers, C., and Karypis, G. (2015). A comprehensive survey of neighborhood-based recommendation methods. In Recommender Systems Handbook, pages 37–76.

O. F. . J. R. van Amersfoort. (2015). Variational recurrent auto-encoders. inICLR, 2015.

P. Alexandru Chirita, W. Nejdl, C. Zamfir (2005). Preventing Shilling Attacks in Online Recommender Systems. Proceedings of the 7th annual ACM international workshop on Web information and data management.

P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, vol. 11, pp. 3371–3408, 2010.

Palanivel, K., and R. Siavkumar. (2010). Fuzzy multi criteria decision-making approach for collaborative recommender systems. International Journal of Computer Theory and Engineering 2, no. 1, pp 57-63

Papageorgiou, E., Georgoulas, G., Stylios, C., Nikiforidis, G. and Groumpos, P., (2006). Combining fuzzy cognitive maps with support vector machines for bladder tumor grading. In International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (pp. 515-523). Springer Berlin Heidelberg.

Papageorgiou, E.I. & Poczeta, K., (2015). Application of fuzzy cognitive maps to electricity consumption prediction. In 2015 Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS) held jointly with 2015 5th World Conference on Soft Computing (WConSC). IEEE, pp. 1–6.

Papageorgiou, E.I. & Salmeron, J.L., (2013). A review of fuzzy cognitive maps research during the last decade. IEEE Transactions on Fuzzy Systems, 21(1), pp.66–79.

Papageorgiou, E.I., Poczeta, K. & Laspidou, C., (2016). Hybrid model for water demand prediction based on fuzzy cognitive maps and artificial neural networks. In 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). IEEE, pp. 1523–1530.

Park, D.H., Kim, H.K., Choi, I.Y., Kim, J.K. (2012). A literature review and classification of recommender systems research. Expert Systems with Applications 39 (11) , pp. 10059-10072.

Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In The adaptive web (pp. 325-341). Springer, Berlin, Heidelberg.

Pham, M.C., Cao, Y., Klamma, R., Jarke, M. (2011). A clustering approach for collaborative filtering recommendation using social network analysis. Journal of Universal Computer Science 17 (4) , pp. 583-604.

Porcel, C., Herrera-Viedma, E. (2010). Dealing with incomplete information in a fuzzy linguistic recommender system to disseminate information in university digital libraries. Knowledge-Based Systems 23 (1), pp. 32-39.

Porcel, C., Tejeda-Lorente, A., Martínez, M.A., Herrera-Viedma, E. (2012). A hybrid recommender system for the selective dissemination of research resources in a technology transfer office. Information Sciences 184 (1) , pp. 1-19.

Pu, P., Chen, L., Hu, R. (2012). Evaluating recommender systems from the user's perspective: Survey of the state of the art. User Modeling and User-Adapted Interaction 22 (4-5) , pp. 317-355.

Quercia, D., Lathia, N., Calabrese, F., Di Lorenzo, G., Crowcroft, J. (2010). Recommending social events from mobile phone location data. Proceedings - IEEE International Conference on Data Mining, ICDM , art. no. 5694070 , pp. 971-976.

R. Salakhutdinov, A. Mnih, and G. Hinton. (2007). Restricted Boltzmann machines for collaborative filtering. In Proc. ICML'07. 791–798.

Rashid, A.M., Albert, I., Cosley, D., Lam, S.K., McNee, S.M., Konstan, J.A., Riedl, J. (2002). Getting to know you: Learning new user preferences in recommender systems. International Conference on Intelligent User Interfaces, Proceedings IUI , pp. 127-134.

Rendle, S., Freudenthaler, C., Gantner, Z. and Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence, pp. 452-461

Rendle, S., Freudenthaler, C., Schmidt-Thieme, L. (2010). Factorizing personalized Markov chains for next-basket recommendation. Proceedings of the 19th International Conference on World Wide Web, WWW '10 , pp. 811-820.

Rendle, S., Schmidt-Thieme, L. (2010). Pairwise interaction tensor factorization for personalized tag recommendation. WSDM 2010 - Proceedings of the 3rd ACM International Conference on Web Search and Data Mining , pp. 81-90.

Ruslan Salakhutdinov and Andriy Mnih. (2007). Probabilistic Matrix Factorization. In Proc. NIPS'07.

Ruslan Salakhutdinov and Andriy Mnih. (2011). Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo. In Proc. ICML'11.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Berg, A. C.(2015). Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 115(3), 211-252

S. Hochreiter and J. Schmidhuber. (1997). Long short-term memory. Neuralcomputation 9, 8 (1997), 1735–1780.

S. P. Chatzis and D. Kosmopoulos. (2011). A Variational Bayesian Methodology for Hidden Markov Models utilizing Student's-t Mixtures. Pattern Recognition, vol. 44, no. 2, pp. 295–306.

S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. (2009). BPR:Bayesian personalized ranking from implicit feedback," in Proc. UAI'09,2009, pp. 452–461.

Salakhutdinov, R. and Mnih, A. (2011). Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo. In Proc. ICML'11.

Schafer, B., Konstan, J. and Riedl. J., (1999). Recommender systems in e-commerce. In Proc. 1st ACM Conference on Electronic Commerce, EC '99. ACM, New York,NY, USA, 158–166.

Šter, B. and Dobnikar, A., (1996). Neural networks in medical diagnosis: Comparison with other methods. In International Conference on Engineering Applications of Neural Networks, pp. 427-430, 1996.

Salakhutdinov, R., Mnih, A. and Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering. In Proceedings of the 24th international conference on Machine learning, pp. 791-798.

Sarwar, Badrul, George Karypis, Joseph Konstan, and John Riedl, (2001). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web, ACM, pp. 285-295.

Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S. (2007). Collaborative filtering recommender systems. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 4321 LNCS , pp. 291-324.

Schein, A.I., Popescul, A., Ungar, L.H., Pennock, D.M. (2002). Methods and metrics for cold-start recommendations. SIGIR Forum (ACM Special Interest Group on Information Retrieval) , pp. 253-260.

Sean M. McNee, John Riedl, Joseph A. Konstan. (2006). Making Recommendations Better: An Analytic Model for Human- Recommender Interaction. In Proceedings of the CHI '06 Extended Abstracts on Human Factors in Computing Systems

Serrano-Guerrero, J., Herrera-Viedma, E., Olivas, J.A., Cerezo, A., Romero, F.P. (2011). A google wave-based fuzzy recommender system to disseminate information in University Digital Libraries 2.0. Information Sciences 181 (9) , pp. 1503-1516.

Seung-Min Han, Mohammad Mehedi Hassan, Chang-Woo Yoon, Hyun-Woo Lee and Eui-Nam Huh (2009). Efficient Service Recommendation System for Cloud Computing Market. Volume 63 of the series Communications in Computer and Information Science pp 117-124.

Shang, M.-S., Zhang, Z.-K., Zhou, T., Zhang, Y.-C. (2010). Collaborative filtering with diffusion-based similarity on tripartite graphs. Physica A: Statistical Mechanics and its Applications 389 (6), pp. 1259-1264.

Shepitsen, A., Gemmell, J., Mobasher, B., Burke, R. (2008). Personalized recommendation in social tagging systems using hierarchical clustering. RecSys'08: Proceedings of the 2008 ACM Conference on Recommender Systems , pp. 259-266.

Shin, K.-S., Lee, T.S. & Kim, H., (2005). An application of support vector machines in bankruptcy prediction model. Expert Systems with Applications, 28(1), pp.127–135.

Shinichi Nakajima and Masashi Sugiyama. (2011). Theoretical Analysis of Bayesian Matrix Factorization. J. Machine Learning Research 12 (2011), 2583–2648.

Shiva Nadi, Mohammad Hossein, and Ayoub Bagheri, (2011). A Hybrid Recommender System for Dynamic Web Users. International Journal Multimedia and Image Processing (IJMIP), Volume 1, Issue 1

Smith, J.W. et al., (1988). Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus. Proceedings of the Annual Symposium on Computer Application in Medical Care, p.261.

Song, Y., Zhang, L., Giles, C.L. (2011). Automatic tag recommendation algorithms for social recommender systems. ACM Transactions on the Web 5 (1) , art. no. 4.

Sotirios P. Chatzis. (2012). A Coupled Indian Buet Process Model for Collaborative Filtering. In Journal of Machine Learning Research: Workshop and Conference Proceedings, Vol. 25: ACML 2012. 65–79.

Sotirios P. Chatzis. (2013). Nonparametric Bayesian Multitask Collaborative Filtering. In Proc. CIKM'13.

Steck, H. (2010). Training and testing of recommender systems on data missing not at random. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining , pp. 713-722.

Stephan Speigel, Jerome Kunegis, Fang Li (2009). Hydra: A Hybrid Recommender System", In Proceedings of the 1st ACM international workshop on Complex networks meet information & knowledge management, pp. 75-80, New York, USD

Su, J.-H., Wang, B.-W., Hsiao, C.-Y., Tseng, V.S. (2010). Personalized rough-set-based recommendation by integrating multiple contents and collaborative information. Information Sciences 180 (1) , pp. 113-131.

Suksom, N., Buranarach, M., Thein, Y.M., Supnithi, T., and Netisopakul, P. (2010). A knowledge-based framework for development of personalized food recommender system. In Proc. of the 5th Int. Conf. on Knowledge, Information and Creativity Support Sys- tems.

Symeonidis, P., Nanopoulos, A., Manolopoulos, Y. (2010). A unified framework for providing recommendations in social tagging systems based on ternary semantic analysis. IEEE Transactions on Knowledge and Data Engineering 22 (2) , art. no. 4815246 , pp. 179-192.

Tan, Y. K., Xu, X., & Liu, Y. (2016). Improved recurrent neural networks for session-based recommendations. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, pp. 17-22

Team, T. T. D., Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., ... & Belopolsky, A. (2016). Theano: A Python framework for fast computation of mathematical expressions. arXiv preprint arXiv:1605.02688.

Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito, Markus Zanker (2012). Linked Open Data to support Content-based Recommender Systems. Proceedings of the 8th International Conference on Semantic Systems.

Tsoukiàs, Alexis, Nikolaos Matsatsinis, and Kleanthi Lakiotaki, (2011). Multi-criteria user modeling in recommender systems. Journal of Intelligence Systems IEEE, vol. 26, pp 64-76

U. Rehman, Z. Hussain (2011). Towards Multi-criteria Cloud Service Selection. Fifth International Conference, Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS).

Vargas, S., Castells, P. (2011). Rank and relevance in novelty and diversity metrics for recommender systems. RecSys'11 - Proceedings of the 5th ACM Conference on Recommender Systems , pp. 109-116.

Walter, F.E., Battiston, S., Schweitzer, F. (2008). A model of a trust-based recommendation system on a social network. Autonomous Agents and Multi-Agent Systems 16 (1) , pp. 57-74.

Wang, C., Blei, D.M. (2011). Collaborative topic modeling for recommending scientific articles. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 448-456.

Wang, H., Wang, N., and Yeung, D.Y. (2015). Collaborative deep learning for recommender systems. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, pages 1235-1244

Wang, J., De Vries, A.P., Reinders, M.J.T. (2006). Unifying user-based and item-based collaborative filtering approaches by similarity fusion. Proceedings of the Twenty-Ninth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval 2006 , pp. 501-508.

Wei, D., Zhou, T., Cimini, G., Wu, P., Liu, W., Zhang, Y.-C. (2011). Effective mechanism for social recommendation of news. Physica A: Statistical Mechanics and its Applications 390 (11) , pp. 2117-2126.

Wen Y. Chen, Jon C. Chu, Junyi Luan, Hongjie Bai, Yi Wang, and Edward Y.Chang. (2009). Collaborative filtering for Orkut communities: discovery of userlatent behavior. In Proc. WWW'09. 681–690.

X. Glorot and Y. Bengio. (2010). Understanding the difficulty of training deepfeedforward neural networks," in Proc. AISTATS.

Xiang, L., Yuan, Q., Zhao, S., Chen, L., Zhang, X., Yang, Q., and Sun, J. (2010). Temporal recommendation on graphs via long- and short-term preference fusion. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Min- ing, Washington, DC, USA, July 25-28, 2010, pages 723–732.

Xiang, L., Yuan, Q., Zhao, S., Chen, L., Zhang, X., Yang, Q., Sun, J. (2010). Temporal recommendation on graphs via long- and short-term preference fusion. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining , pp. 723-731.

Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu. (2013). Advances in optimizing recurrent networks. In Proc. ICASSP. 8624–8628.

Y. Koren. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proc.14th ACM SIGKDD. Las Vegas, Nevada, USA.

Y. LeCun, Y. Bengio, and G. Hinton. (2015). Deep learning. Nature 512 (2015),436–444.

Y. Zhang, H. Dai, C. Xu, J. Feng, T. Wang, J. Bian, B. Wang, and T.-Y. Liu. (2014). Sequential click prediction for sponsored search with recurrent neural networks. In Proc. AAAI-14. 1369–1375.

Yang, J., Wang, Z., and Zhang, X. (2015). An ibeacon-based indoor positioning systems for hospitals. Inter- national Journal of Smart Home, 9(7):161–168.

Yilei Zhang, Zibin Zheng and Michael R. Lyu (2011). Exploring Latent Features for Memory-Based QoS Prediction in Cloud Computing. 30th IEEE International Symposium on Reliable Distributed Systems.

Yuan, W., Guan, D., Lee, Y.-K., Lee, S., Hur, S.J. (2010). Improved trust-aware recommender system using small-worldness of trust networks. Knowledge-Based Systems 23 (3) , pp. 232-238.

Zhan, J., Hsieh, C.-L., Wang, I.-C., Hsu, T.-S., Liau, C.-J., Wang, D.-W. (2010). Privacy-preserving collaborative recommender systems. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews 40 (4) , art. no. 5411745 , pp. 472-476.

Zhang, Y., Dai, H., Xu, C., Feng, J., Wang, T., Bian, J., Wang, B. and Liu, T.Y. (2014). Sequential click prediction for sponsored search with recurrent neural networks. arXiv preprint arXiv:1404.5772.

Zhang, Z.-K., Zhou, T., Zhang, Y.-C. (2010). Personalized recommendation via integrated diffusion on user-item-tag tripartite graphs. Physica A: Statistical Mechanics and its Applications 389 (1) , pp. 179-186.

Zheng, N., Li, Q. (2011). A recommender system based on tag and time information for social tagging systems. Expert Systems with Applications 38 (4) , pp. 4575-4587.

Zheng, Y., Zhang, L., Ma, Z., Xie, X., Ma, W.-Y. (2011). Recommending friends and locations based on individual location history. ACM Transactions on the Web 5 (1) , art. no. 5.

Zhou W, Wen J, Koh YS, Xiong Q, Gao M, et al. (2015). Shilling Attacks Detection in Recommender Systems Based on Target Item Analysis. Shilling Attacks Detection in Recommender Systems Based on Target Item Analysis.

Zhoua, T., Kuscsik, Z., Liu, J.-G., Medo, M., Wakeling, J.R., Zhang, Y.-C. (2010). Solving the apparent diversity-accuracy dilemma of recommender systems. Proceedings of the National Academy of Sciences of the United States of America. Volume 107, Issue 10, Pages 4511-4515.

Zibin Zheng, Yilei Zhang and Michael R. Lyu (2010). CloudRank: A QoS-Driven Component Ranking Framework for Cloud Computing. 29th IEEE International Symposium on Reliable Distributed Systems.

Adomavicius, G., & Tuzhilin, A. (2015). Context-aware recommender systems. In Recommender systems handbook (pp. 191-226). Springer US.

Zhu, T., Li, G., Zhou, W., Xiong, P., & Yuan, C. (2014, May). Deferentially private tagging recommendation based on topic model. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 557-568). Springer, Cham.

Mo, Y., Chen, J., Xie, X., Luo, C., & Yang, L. T. (2014). Cloud-based mobile multimedia recommendation system with user behavior information. IEEE Systems Journal, 8(1), 184-193.

Roy, S. B., Thirumuruganathan, S., Amer-Yahia, S., Das, G., & Yu, C. (2014, March). Exploiting group recommendation functions for flexible preferences. In Data Engineering (ICDE), 2014 IEEE 30th International Conference on (pp. 412-423). IEEE.

Amatriain, X., & Pujol, J. M. (2015). Data mining methods for recommender systems. In Recommender Systems Handbook(pp. 227-262). Springer US.

Adomavicius, G., & Kwon, Y. (2015). Multi-criteria recommender systems. In Recommender Systems Handbook(pp. 847-880). Springer US.

Nilashi, M., bin Ibrahim, O., Ithnin, N., & Sarmin, N. H. (2015). A multi-criteria collaborative filtering recommender system for the tourism domain using Expectation Maximization (EM) and PCA–ANFIS. Electronic Commerce Research and Applications, 14(6), 542-562.

Vera-del-Campo, J., Pegueroles, J., Hernández-Serrano, J., & Soriano, M. (2014). DocCloud: A document recommender system on cloud computing with plausible deniability. Information Sciences, 258, 387-402.

Meymandpour, R., & Davis, J. G. (2015, January). Enhancing Recommender Systems using Linked Open Data-based semantic analysis of items. In Proceedings of the 3rd australasian web conference (AWC 2015) (Vol. 27, p. 30).

Harman, J. L., O'Donovan, J., Abdelzaher, T., & Gonzalez, C. (2014, October). Dynamics of human trust in recommender systems. In Proceedings of the 8th ACM Conference on Recommender systems (pp. 305-308). ACM.