Department of Production & Management Engineering

Technical University of Crete

École doctoral décision informatique mathématiques organization

Université Paris Dauphine

An integrated Recommender System based on Multi-Criteria Decision Analysis and Data Analysis methods: Methodology, implementation and evaluation

by

Kleanthi Lakiotaki

Submitted for the partial fulfillment of the requirements for the degree of
Doctor of Philosophy

December 2010

Declarations

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Parts of the thesis have been published in academic journals or conference proceedings. Please cite as appropriate when referring to this text.

This thesis is approved by:

1. Nikolaos Matsatsinis, Professor (*advisor in Technical University of Crete)*

2. Alexis Tsoukiàs, Directeur de recherche CNRS *(advisor in Université Paris-Dauphine)*

3. Anastasios Doulamis, Assistant Professor *(advisor in Technical University of Crete)*

4. Vangelis Paschos, Professor in Université Paris-Dauphine

5. Maro Vlachopoulou, Professor in University of Macedonia

6. Vassiliki, Manthou – Fragopoulou, Professor in University of Macedonia

7. Athanasios Migdalas, Professor in Aristotle University of Thessaloniki

Contents

Acknowledgments

The completion of this thesis leaves me with an intermixture of thoughts and emotions. The first and surely the most sincere feeling is that of gratefulness to all the people that directly or not, helped me go through the turbulent times!

First, I would like to thank my supervisor, Professor Nikolaos Matsatsinis, for his guidance and support in dealing with my thesis, for letting me run with my own ideas and allowing them to flourish and also for my three-year scholarship.

I also thank Professor Alexis Tsoukias for his insightful remarks and for his philosophical spirit that inspired me all the way through my research.

Many thanks to Professor Tasos Doulamis for our endless scientific discussions and for his inexhaustible energy that generously spread throughout our lab.

I would also like to thank all the members of my examining committee for their time and consideration and of course for their acceptance to participate.

I am indebted to Professors Michael Doubos and Evangelos Grigoroudis for always being there to answer my questions with great willingness and also to Professor Nikos Vlassis that introduced me the global k-means algorithm which I incorporated into this work.

I would never forget to thank all my wonderful colleagues and friends at the Technical University of Crete. In particular, I would like to thank Pavlos Delias (now Dr. and father!) for his support and friendship all this time and for his unlimited patience…

Many thanks to Stelios Tsafarakis for our countless discussions about science and more…

Vassilis Fortsas helped in various ways with his expertise in computer equipment and by always being friendly and supporting.

It is impossible to describe in a few words my gratitude to Panagiotis Kontogiannis for his friendship and support in difficult times…Without him "locking" me at the computer rooms I never would have made it on time!

Many thanks to my office mate Lefteris that we shared many hours together and he has now joined the army (I wonder why he left…).

And because in most cases there is always the "person behind the scene", who does a wonderful job and takes care of everything and everybody, I would never forget to express my appreciation to Lia Krasadaki and wish her to get her PhD really really soon!!!

All the members of ERGA.S.Y.A have been more than helpful and friendly to me and for this I deeply thank them.

My almost four years at the Technical University of Crete coincided with my growing interest in Computer Science. In this respect, I would like to thank my dearest friend Dr. Vangelis Sakkalis, with whom I discussed these topics quite frequently. I am pleased that his hard work has been finally recognized and I wish him all the best for his new position as Researcher.

Since our problems always weighed less in our hearts once shared with each other I would like to thank at this point my closest friends, Rena, Irene1, Irene2, Irene3, Irene4, Dora, Glykeria, Artzi and Despoina for always listening to my complains and encouraging me to keep walking! I promise to never say no again to future coffee invitations!!

My heartfelt thanks to Christos for our philosophical discussions, which of course never led to anywhere, but they were more than entertaining and enjoyable!

Maybe my most deep thanks go to my parents for all their emotional and financial support, their understanding and love. This thesis is therefore dedicated to them.

This work is part of the 03ED375 research project, implemented within the framework of the "Reinforcement Program of Human Research Manpower" (PENED) and co-financed by National and Community Funds (75% from E.U.-

European Social Fund and 25% from the Greek Ministry of Development-General Secretariat of Research and Technology).

Short Curriculum Vitae

Kleanthi Lakiotaki studied Physics at the University of Crete and she also completed an MRes course at the same University. During her studies there, she was an undergraduate and postgraduate fellow at the Foundation of Research and Technology (FORTH). Afterwards, she was a Marie Curie research fellow at Imperial College London. During her PhD studies in information and decision sciences at the Production & Management Engineering Department of Technical University of Crete and at the Université Paris Dauphine under a cotutelle framework she was funded by the General Secretariat of Research and Technology in Greece. She has been author or co-author of 8 journal articles, 1 book section and several peer reviewed conference articles. Her research interests include Recommender Systems, Business Intelligence, Personalization Technologies, Decision Support Systems and Multi-Criteria Decision Analysis.

Contact her at the Dept. of Production and Management Engineering, Technical University of Crete, GR 73100, Chania, Crete, Greece; klio@ergasya.tuc.gr or kliolak@gmail.com

Τίτλος και περίληψη στα Ελληνικά

Τίτλος: *"Ανάπτυξη ενός συστήματος ευφυών πρακτόρων για την αναζήτηση και ανάλυση πληροφοριών στο διαδίκτυο, την αυτοματοποιημένη ανάπτυξη ερωτηματολογίων και διενέργειας ερευνών αγοράς βασιζόμενου στην μοντελοποίηση των προτιμήσεων του χρήστη μέσω μεθόδων ανάλυσης δεδομένων και πολυκριτήριας ανάλυσης"*

Περίληψη: Με τη συνεχή αύξηση της διαθέσιμης πληροφορίας στον παγκόσμιο ιστό, την ποικιλομορφία των χρηστών του και την πολυπλοκότητα των διαδικτυακών εφαρμογών, οι ερευνητές άρχισαν να αμφισβητούν την γενική προσέγγιση του «one size fits all». Έχει νόημα μια εφαρμογή ηλεκτρονικού εμπορίου, για παράδειγμα, να παρουσιάσει τα ίδια προϊόντα σε χρήστες του διαδικτύου με πολύ διαφορετικές προτιμήσεις;

Για την αντιμετώπιση αυτού του είδους των ερωτημάτων, οι ερευνητές του χώρου ξεκίνησαν την ανάπτυξη διαδικτυακών συστημάτων που έχουν τη δυνατότητα να προσαρμόζουν την εμφάνισή και τη συμπεριφορά τους σε κάθε μεμονωμένο χρήστη ή ομάδα χρηστών. Τα Συστήματα Συστάσεων (Recommender Systems), ένα εξελιγμένο είδος των λεγόμενων προσαρμοστικών συστημάτων του παγκοσμίου ιστού (adaptive web based systems), σκοπό έχουν να στηρίξουν την αναζήτηση και την περιήγηση των χρηστών του διαδικτύου εντοπίζοντας προϊόντα ή υπηρεσίες ενδιαφέροντα για τον εκάστοτε χρήστη του συστήματος, από μια πληθώρα προσφερομένων. Αυτά τα συστήματα που αναπτύσσονται περίπου εδώ και 20 χρόνια, παρά την εκθετική ανάπτυξη τους θεωρούνται ακόμη σε πρώιμο στάδιο ανάπτυξης από ερευνητική άποψη. Αυτό σημαίνει ότι, οι διάφορες πτυχές των εν λόγω συστημάτων παραμένουν ανεξερεύνητες.

Η παρούσα διατριβή στοχεύει στην παροχή νέων, πρωτότυπων ιδεών για την έρευνα του προβλήματος συστάσεων, μέσω της εισαγωγής μεθοδολογιών και τεχνικών από τον ευρύτερο τομέα της Πολυκριτήριας Ανάλυσης Αποφάσεων στο ερευνητικό πεδίο των Συστημάτων Συστάσεων. Στην παρούσα διατριβή προτείνεται ένα υβριδικό μεθοδολογικό πλαίσιο, το οποίο συγχωνεύει τεχνικές τόσο από το χώρο της Πολυκριτήριας Ανάλυσης Αποφάσεων, όσο και από αυτόν των Συστημάτων Συστάσεων. Το μεθοδολογικό αυτό πλαίσιο, το οποίο παρουσιάζεται και αναλύονται λεπτομερώς, και αποτελεί ταυτόχρονα την κύρια συνεισφορά της

παρούσας διατριβής, εφαρμόζεται μέσω του UTARec, ενός συστήματος που ενσωματώνει την προτεινόμενη μεθοδολογία και επιδεικνύει την λειτουργία του. Η συμβολή της διατριβής αυτής οφείλεται κυρίως στο δυναμικό το οποίο αναπτύσσεται μέσω του προτεινόμενου υβριδικού μεθοδολογικού πλαισίου και μπορεί να ταξινομηθεί με βάση τους διάφορους τομείς που εν δυνάμει επωφελούνται από τα αποτελέσματα αυτής της διατριβής, όπως η μοντελοποίηση χρηστών (User Modeling), τα Συστήματα Συστάσεων (Recommender Systems), η Πολυκριτήρια Ανάλυση Αποφάσεων και το ηλεκτρονικό μάρκετινγκ (e-marketing). θεματική αναφορά των επιμέρους συνιστωσών που θεμελιώνουν τη συνολική συμβολή της εργασίας αυτής μπορεί να βρεθεί στο υποκεφάλαιο 1.2.

Η παρούσα διατριβή υποστηρίζει ότι η χρήση μεθοδολογιών από το χώρο της Πολυκριτήριας Ανάλυσης Αποφάσεων μπορεί να αποδειχθεί χρήσιμη για την επίλυση προβλημάτων που συχνά συναντώνται στα Συστήματα Συστάσεων. Πιο συγκεκριμένα, ορισμένοι από τους σημαντικότερους περιορισμούς της Συστημάτων Συστάσεων που βασίζονται στη συνεργατική διήθηση, όπως το ονομαζόμενο πρόβλημα της «καθυστερημένης εκκίνησης», ή αυξημένη διασπορά των δεδομένων, ή το πρόβλημα του ασυνήθιστου αξιολογητή, περιορίζονται σημαντικά στην περίπτωση των συστημάτων συστάσεων τύπου UTARec. Επιπρόσθετα, βασικά προβλήματα των Συστημάτων Συστάσεων που βασίζονται στο περιεχόμενο, όπως η εξάρτηση τους από την εξαγωγή χαρακτηριστικών επίσης περιορίζονται σημαντικά. Αναλυτικές λεπτομέρειες για το πώς τα προβλήματα αυτά αντιμετωπίζονται μπορούν να βρεθούν σε στα επιμέρους συμπεράσματα των κεφαλαίων, ενώ συνοψίζονται επίσης στο υποκεφάλαιο 6.1. Αυτή η διατριβή ολοκληρώνεται, με αναφορά στις πιθανές μελλοντικές πτυχές της, καθώς και σε διάφορες ιδέες για περαιτέρω έρευνα.

Abstract in English

With the growth of the available information on the Web, the diversity of its users and the complexity of Web applications, researchers started to question this generic approach of “one size fits all”. Does it make sense for an e-commerce Web site for example, to present the same products to internet users with widely diverse preferences?

To address this kind of questions, researchers started developing adaptive Web systems that tailored their appearance and behavior to each individual user or user group. Recommender systems, a sophisticated type of adaptive Web system, assist search and browsing based information tasks, by recommending items that seem most relevant to users’ interests and might otherwise be missed due to information overload. These systems are being developed for about 20 years now and despite their exponential growth, they are still considered in their infancy from a research point of view. This means that yet, several aspects of these systems are to be explored.

This thesis aims at providing new insights to the recommendation problem by introducing the exploitation of methodologies and techniques from the greater field of Multiple Criteria Decision Analysis (MCDA) to the Recommender Systems research field. A hybrid methodological framework that merges techniques both from the Multiple Criteria Decision Analysis and the Recommender Systems research areas, is described and analyzed in details. This framework, which also constitutes the major outcome of this thesis, is implemented via the UTARec, a system that incorporates the proposed methodology and thus demonstrates its performance. The contribution of this work lies mainly on the potentiality of the proposed hybrid methodological framework and can be divided based on the various disciplines that are benefitted from the results of this thesis, such as User Modeling, Recommender Systems, Multiple Criteria Decision Analysis and e-marketing. A thematic reference of the individual components that build the overall contribution of this work can be found in section 1.2.

It is advocated in this thesis that methodologies from the MCDA field can be proved helpful in solving common problems of Recommender Systems. In particular, some of the major shortcomings of current collaborative filtering

Recommender Systems such as the so called “cold start” problem, the data sparseness or the unusual rater problem, are limited in the case of UTARec type Recommender Systems. Moreover, common problems of existing content based Recommender Systems such as the feature extraction dependence are also addressed at some point by systems designed according to the proposed methodology. Analytical details on how these problems are treated can be found throughout the individual chapter conclusions, while they are also summarized in 6.1. This thesis ends, with a reference on possible future aspects of this work, providing thus ideas for further research.

Titre et résumé en Français

Titre: *"Un système intégré de recommandation basé sur la multi-Criteria Decision Analysis et des méthodes d'analyse de données: méthodologie, mise en œuvre et l'évaluation"*

Résumé: Avec la croissance de l'information disponible sur le Web, la diversité de ses utilisateurs et la complexité des applications du Web, les chercheurs ont commencé à douter cette approche générique de "one size fits all". Est-il judicieux pour un site e-commerce du Web par exemple, de présenter les mêmes produits aux utilisateurs de l'internet qui ont des préférences très diverses?

Pour aborder ce type de questions, les chercheurs ont commencé à développer des systèmes adaptatifs du Web qui adaptemt à leur apparence et au comportement pour chaque utilisateur ou groupe d'utilisateurs. Les systèmes de recommandation facilitent la recherche et la navigation des tâches basées sur l'information, en recommandant des articles qui semblent les plus pertinentes aux intérêts des utilisateurs et pourraient autrement être manquées pour cause de la surcharge de l' information. Ces systèmes sont développés depuis environ 20 ans et malgré de leur croissance exponentielle, ils sont considérés qu' ils sont encore en stade précoce d'un point de vue de la recherche. Cela signifie que pour le moment, plusieurs aspects de ces systèmes peuvent explorer.

Cette thèse vise à fournir de nouvelles perspectives dans le domaine des systèmes de recommandations en introduisant de l'exploitation des méthodes et techniques du plus grand champ de Multiple Criteria Decision Analysis (MCDA) vers la recherche des Systèmes de recommandation. Un cadre méthodologique hybride qui fusionne les deux techniques de l'analyse multicritère à la décision domaine de la recherche et le Recommender Systems domaine de la recherche est décrit et analysé dans les détails. Ce dernier, qui constitue également le principal résultat de cette thèse, est mis en œuvre par UTARec, un système qui incorpore le cadre proposé et démontre ses performances. La contribution de ce travail réside principalement dans la potentialité du cadre méthodologique de l' hybride proposée et peut être divisée sur la base des diverses disciplines qui sont bénéficié des résultats de cette thèse, comme User Modeling, Recommender

Systems, Multiple Criteria Decision Analysis and e - marketing. Une référence thématique de chacun des composants qui renforcent la contribution globale de cette thèse se trouve au point 1.2.

Il est préconisé dans cette thèse que les méthodes sur le terrain MCDA peut être prouvé qu' elles sont utiles pour résoudre les problèmes communs des systèmes de recommandation. En particulier, certaines des faiblesses principales de l'actuel filtrage collaboratif Recommender systèmes tels que le soi-disant "cold start" problème, la rareté des données ou le problème de l'utilisateur inhabituel, sont limités dans le cas de UTARec Recommender Type Systems. En outre, les problèmes communs de contenus existants basés aux Recommender Systems comme la dépendance à l'extraction de caractéristiques, sont également adressées à un certain moment par des systèmes conçus conformément à la méthodologie proposée. Des détails analytiques sur la façon dont ces problèmes se sont trouvés dans les conclusions du chapitre individuel, ils sont également résumés au point 6.1. Cette thèse se termine, avec une référence sur les futures aspects possibles de ce travail, en fournissant ainsi des idées pour d'autres recherches.

List of publications

- **1**. K.Lakiotaki, N. F. Matsatsinis, and A.Tsoukiàs, “Multi-Criteria User Modeling in Recommender Systems”, ***IEEE Intelligent Systems*** (accepted)
- **2**. K. Lakiotaki, P. Delias, V. Sakkalis and N. F. Matsatsinis, “User Profiling based on Multi-criteria Analysis: The role of Utility Functions”, ***Operational Research: An International Journal***, 9(1),3-16, (2009)
- **3.** K. Lakiotaki, N. F. Matsatsinis, “Analyzing User Modeling in a Multi-Criteria Movie Recommender System”, ***ACM Recommender Systems 2009***, (1st International Workshop on Recommendation-based Industrial Applications) October 22-25, New York, USA
- **4.** K. Lakiotaki, S. Tsafarakis, N. F. Matsatsinis, “UTA-REC: A Recommender system based on Multiple Criteria Analysis”, ***ACM Recommender Systems 2008***, October 23-25, Lausanne, Switcherland
- **5.** S. Tsafarakis, K. Lakiotaki and N. Matsatsinis, “Applications of MCDA in Marketing and e-Commerce”, ***Handbook of Multicriteria Analysis***, Springer (to appear)
- **6.** N. F. Matsatsinis, K. Lakiotaki, P. Delias, “A System based on Multiple Criteria Analysis for Scientific Paper Recommendation”, ***11th Pan-Hellenic Conference on Informatics*** *(PCI 2007)*, Patras, Greece
- **7.** K. Lakiotaki, S. Tsafarakis, N. F. Matsatsinis, “Leveraging Recommender Systems: The use of Multiple Criteria Analysis to enhance Collaborative Filtering Algorithm”, ***6η Συνάντηση Πολυκριτήριας Ανάλυσης Αποφάσεων***, 30 Σεπτεμβρίου 2008, Χανιά, Κρήτη (oral presentation)
- **8.** K. Lakiotaki, N. F. Matsatsinis, “Learning Customer Profiles: Methodology and Implementation”, ***67th Meeting of the EWG on MCDA***, April 3-5, 2008, Rovaniemi, Finland (discussion paper)
- **9.** N.F. Matsatsinis, K. Lakiotaki, P. Delias, “Clustering customers according to their preferences”, ***66th Meeting of the EWG on MCDA***, October 18-20, 2007, Marrakech, Morocco (discussion paper)

List of figures

List of tables

1 Introduction

Contents

1.1 Motivation

The world wide web (or simply "the Web") had an estimated 1,6 billion users in March 2009 (http://www.internetworldstats.com/stats.htm), an amount that corresponds to the 23,8% of the total population on earth. Needless to mention, that the www is accessed by people of essentially all possible backgrounds. Each of these users has a goal in mind, whether it is trying to book a flight, search for information on a research topic, or just spend aimlessly a few hours. Different users also have different knowledge, interests, abilities, learning styles, and preferences regarding information presentation. In the first few years after its inception, the Web was the same for everyone. Web sites presented the same information and the same links to all visitors, regardless of their goals and prior knowledge. A query to a Web search engine or catalog produced the same result for all users, irrespective of their underlying interests and information needs.

With the growth of the available information on the Web, the diversity of its users and the complexity of Web applications, researchers started to question this generic approach of "one size fits all". Does it make sense for an e-commerce Web site for example, to present the same products to internet users with widely diverse preferences?

To address this kind of questions, researchers started developing adaptive Web systems that tailored their appearance and behavior to each individual user or user group. Web based adaptive systems were designed for different usage contexts and explored different kinds of personalization. Since early stages of their development, adaptive systems have penetrated in various fields and serve numerous user needs. For instance, adaptive hypermedia systems, mainly applied in education, tailor page content to the learner's needs, adaptive search systems promote items in result lists that are considered more relevant to the user's interests and needs than others (P. Brusilovsky, A. Kobsa, *et al.* 2007).

Adaptive filtering and Recommender systems assist search and browsing based information, by recommending items that seem most relevant to users' interests and might otherwise be missed due to information overload. To support these kinds of personalization, adaptive systems collect data about their users by implicitly observing their interaction and explicitly requesting direct input from them, and they build user models (aka "profiles") that enable them to deliver different information to users.

Year after year, the growing demands on personalization as well as the success of early adaptive Web systems resulted in progressively more advanced systems. Web personalization has grown into a large research field that attracts scientists from different communities such as, user modeling, machine learning, natural language generation and recognition, information retrieval, intelligent tutoring systems, cognitive science, Web-based education and other.

Personalization is considered a key issue in designing adaptive Web systems which mainly adapt techniques from user modeling. As the field of the adaptive Web has reached a certain level of maturity, user modeling, adaptation and personalization are considered a joint area of research that deals with any aspect of systems that acquire information about a user (or group of users) so as to be able to adapt their behavior to that user or group. Meanwhile, the volume of knowledge and experience collected in this broad field gradually turns the adaptive Web, from an area of pure research into an engineering discipline, where new adaptive systems can be quickly developed by combining known techniques and ideas.

At the same time, the rapid expansion of the Internet has resulted in the genesis of a new market with unique and bottomless opportunities for businesses to explore. Electronic commerce, or e-commerce, has enabled businesses to open up their products and services to a massive audience that was once impossible to target. Soon after the explosion of the internet and internet services, the competition between e-businesses became increasingly intense and consumers are mow faced with a myriad of choices. Although this might seem to be beneficial to the consumer, it often leads to overwhelming and frustration.

The idea of applying the principles of adaptive Web to the needs of electronic commerce led to the development of Recommender Systems, the aim of which is to help users to find items that they should appreciate from huge catalogues. Items can be of any type, such as films, music, books, web pages, online news, jokes, restaurants and even lifestyles.

The amalgamation of user modeling, adaptation and personalization, offers numerous opportunities and ideas to Recommender Systems (aka Recommenders or Recommendation Systems). Since their early stages of development in mid-1990, Recommender Systems soon emerged as an independent research field (P. Resnick and H. R. Varian 1997). Nonetheless, the design and development of such systems shares techniques from the broad field of user modeling, adaptation and personalization, which in turn borrows methodologies and techniques from the wider field of Artificial Intelligence and Machine Learning. Numerous research works have been published in the field of Recommender Systems and several e-businesses (amazon.com, ebay.com and others) employ such systems.

As Recommender Systems research matures, new ideas and research approaches emerge, trying to address certain shortcomings of current techniques. Many studies are dedicated in finding ways to reduce user effort, by enhancing the processing of explicit information, some researchers on the other hand, direct their research on broad aspects of Human – Computer Interaction to elicit user preference information, and some others focus explicitly on the design recommendation algorithms with increased recommendation performance. New approaches appear continuously by considering several aspects of Recommender Systems, like contextual Recommenders (G. Adomavicius, R. Sankaranarayanan, *et al.* 2005).

The general comprehension, realization and statement of the lack of research studies in the field of Multi Criteria Recommender Systems (G. Adomavicius and A. Tuzhilin 2005) throughout literature, together with the default multi criteria nature of decision problems involved in Recommenders, laid the foundations of inspiration for this work. Although Multiple Criteria Decision Analysis (MCDA) has been extensively studied in Operational Research (J. Figueira, S. Greco, *et al.* 2005) and has been proved successful in several applications, a relatively small number of studies are encountered in Recommender Systems. This thesis aims at providing new insights to the field of Recommender Systems by introducing the exploitation of methodologies and techniques from the greater field of MCDA to the Recommender Systems research area. A hybrid methodological framework that merges techniques both from the Multiple Criteria Decision Analysis research area and the Recommender Systems research area is described and analyzed in details. The latter, which also constitutes the major outcome of this thesis, is implemented through UTARec, a system that incorporates the proposed framework and demonstrates its performance.

1.2 Contribution

The contribution of this work lies mainly in the potentiality of the proposed hybrid methodological framework and can be divided based on the various disciplines that are benefitted from the results of this thesis, such as User Modeling, Recommender Systems, Multiple Criteria Decision Analysis and e-marketing. A thematic reference of the individual components that build the overall contribution of this thesis follows:

For the Multiple Criteria Decision Analysts:

1) The successful application of the Disaggregation- Aggregation MCDA approach in Recommender Systems provides new insights and research directions for researchers of other MCDA methodologies to consider Recommender System as a potential application field with numerous unexplored aspects.

2) The unified hybrid methodological framework on which the proposed algorithm is build, can also serve as a general framework for the design of additional Multi-Criteria Recommender Systems.

For the Recommender System's area:

3) The UTARec system that has been designed and built to demonstrate the performance of the proposed hybrid approach, now serves as an integrated Multi-Criteria Recommendation System and exhibits increased performance compared to existing approaches commonly used in Recommender Systems.

4) The demand and importance of considering multi-criteria preference information in Recommender Systems is clearly proved by the encouraging results of the comparison analysis.

5) The idea of applying the collaborative filtering philosophy inside user profile groups and not to the entire set of users significantly accelerates the performance of common Recommender Systems techniques.

For the User modeling scientists:

6) The creation of user profiles based on multi-criteria preference information is considered a novel approach of representing user preferences. These user profiles are the result of a sophisticated MCDA approach and represent the general value system under which users think, decide and act. The incorporation of advanced MCDA methods to create user profiles introduces a new, yet insufficiently explored, aspect of the user modeling process.

For the e-marketing researchers:

7) An innovative way of forming market segments, a fundamental step of target marketing, which involves breaking a market into segments and then concentrating marketing efforts on one or a few key segments, emerges from the proposed approach, by studying the population evolution of user profile groups. It is analytically presented in section 5.9 that different groups of similar behavior, regarding a specific aspect of the system, the "movie taste" in the specific application, but it can be equally extended to any other attribute of commercial interest, can and should be approached from a different angle, which may for example represent a different marketing strategy.

8) One of the goals of e-marketing is to increase product sales and customer loyalty by simultaneously preserving customer satisfaction. One way to achieve this goal is by employing personalization technologies. This thesis pioneers the use of advanced methods, like MCDA, in identifying and exploiting user profile groups and proposes a flexible system able to adapt the level of personalization to serve individual needs.

2 Recommender Systems

Contents

2.1 Introduction

"…The Web, they say, is leaving the era of search and entering one of discovery. What's the difference? Search is what you do when you're looking for

something. Discovery is when something wonderful that you didn't know existed, or didn't know how to ask for, finds you". Jeffrey O'Brien at Fortune writes about "The race to create a 'smart' Google".

The advent of the Web and the concomitant increase in information available online, has caused information overload and ignited research in developing systems to overcome this problem. This led to a clear demand for automated methods that locate and retrieve information with respect to users' individual interests. Many conventional human tasks are accomplished by information agents nowadays, and this is proved by the fact that the prefix "e-" is encountered in an increasing number of words every day (e-learning, e-commerce, e.t.c.).

Nevertheless, since information systems are designed by humans, it is more than reasonable, to find inspiration in everyday life and way of thinking. The traditional human process of "word of mouth", for instance, has triggered the development of the collaborative filtering algorithm, which in turn, formed the basis for the development of Recommender Systems. The renaissance that online world is undergoing, laid the foundations for the florescence of personalization technologies in online environments. Thus, Recommender Systems gained much research interest and many research groups focused their work exclusively in the development of such systems.

Recommender Systems have now been an active topic of research for about 20 years. In the early 21st century, many researchers are working on some aspect of Recommender Systems, hundreds of papers have already been published, and several enterprises from newcomers like MyStrands (http://www.strands.com/) and StumbleUpon (http://www.stumbleupon.com/), to titans like Yahoo (http://www.yahoo.com/) and Sun (http://www.sun.com/), have deployed Recommender Systems on the Internet and beyond. In the commercial sector, Amazon (http://www.amazon.com/) realized early how powerful a recommender system could be and to this day remains one of the most representative examples of companies that considerably employ Recommender Systems.

The million-dollar competition sponsored by Netflix (http://www.netflix.com/), an online movie rental company, further indicates the value of Recommender Systems in the electronic commerce age.

"*The effect of Recommender Systems will be one of the most important changes in the next decade*" says University of Minnesota computer science Professor John Riedl, who built along with his colleagues, one of the first recommendation engines in the mid-1990s (P. Resnick, N. Iacovou, *et al.* 1994) and adds: *"…The social web is going to be driven by these systems."*

2.2 Early History & Definitions

Recommender systems have been informally used for many years. A characteristic example is their appearance even in prehistoric days, when our species relied upon informal collaborative filtering (J. Riedl, J. Konstan, et al. 2002). When prehistoric man encountered a new berry, not everyone in the tribe ate it right away. Some would wait to see if the others became sick before trying a new food. If no one became sick, then this acted as a recommendation for eating the berry. If people did become sick then it served as a negative recommendation for the berry in question. This is a rather simplified view of Recommender Systems but accurate nonetheless. To continue the prehistoric example, suppose that one tribe came upon another tribe and shared knowledge. As populations grew and spread, so did knowledge. This is the basic principle behind the collaborative filtering method of Recommender Systems. As technology has advanced, automated systems have been built and other methods employed to make recommendations.

The first formal implementation of Recommender Systems, named Tapestry (D. Goldberg, D. Nichols, et al. 1992), was created in 1992 and it was then that the term "collaborative filtering" was coined to represent the technology behind Recommender Systems. Tapestry relied on the explicit opinion of people from a cohesive community, such as an office group.

Later on, Paul Resnick who is considered a pioneer in the field of Recommender Systems, along with his colleagues, developed a system, named GroupLens, which uses user ratings to recommend news articles to other users

(P. Resnick, N. Iacovou, et al. 1994). Nowadays, Grouplens is a research lab in the Department of Computer Science and Engineering at the University of Minnesota, conducting research in various aspects of Recommender Systems. The GroupLens group has also launched MovieLens (B. J. Dahlen, J. A. Konstan, et al. 1998), a personalized movie recommendation system based on a typical collaborative filtering algorithm that collects movie preferences from users and then groups users with similar tastes. Based on the movie ratings expressed by all the users in a group, it attempts to predict for each individual their opinion on movies they have not yet seen. Since its first performance, Movielens has undergone many improvements as a result of lessons learned from real user interaction experience (J. A. Konstan, J. Riedl, et al. 1998), (B. N. Miller, I. Albert, et al. 2003).

A series of publications developed at the late 20th century, when Recommender Systems were still in their infancy, are gathered in the 3rd issue of Volume 40 of the Communications of the ACM, published in 1997 (P. Resnick and H. R. Varian 1997). There, Grouplens, the collaborative filtering system for Usenet News, is presented in a more abstract, yet integrated approach (J. A. Konstan, B. N. Miller, et al. 1997). Some characteristic Recommender Systems found throughout literature are concisely mentioned below.

Among others, **PHOAKS** (People Helping One Another Know Stuff), an experimental system that addressed the problem of finding relevant, high quality information on the Web, through a collaborative filtering (CF) approach is also considered one of the first, characteristic applications of collaborative filtering (L. Terveen, W. Hill, et al. 1997). PHOAKS works by automatically recognizing, aggregating and redistributing recommendations of Web resources mined from Usenet news messages.

Referral Web (H. Kautz, B. Selman, et al. 1997), is an interactive system for restructuring, visualizing and searching social networks on the Web. Referral Web primarily builds its users' model of its social network by data mining public documents found on the Web and attempts to uncover existing social networks, rather than provide a tool for creating new communities.

Fab (M. Balabanović and Y. Shoham 1997), a recommendation system for the Web, was one of the first systems that introduced the idea of combining both collaborative and content-based filtering systems to eliminate many of the weaknesses of each approach.

Siteseer (J. Rucker and M. J. Polanco 1997) is a web page Recommender System that uses individual's bookmarks and the organization of bookmarks within folders for predicting and recommending relevant pages. Siteseer utilizes each user's bookmarks as an implicit declaration of interest in the underlying content, and the user's grouping behavior (such as the placement of subjects in folders) as an indication of semantic coherency or relevant groupings between subjects. In addition, Siteseer treats folders as a personal classification system which enables it to contextualize recommendations in classes defined by the user.

Various definitions exist on the notion of Recommender Systems throughout literature and the Internet. Some researchers use the concepts, *"Recommender System"*, *"Collaborative Filtering"* and *"Social Filtering"* interchangeably. Others regard "*Recommender System*" as a generic descriptor that represents various recommendation/prediction techniques including collaborative, social, content based filtering and other.

According to Wikipedia, *"Recommender systems form a specific type of information filtering (IF) technique, that attempts to present information items (movies, music, books, news, images, web pages, etc.) that are likely of interest to the user. Typically, a recommender system compares the user's profile to some reference characteristics, and seeks to predict the 'rating' that a user would give to an item they had not yet considered".*

It is may simply stated that a Recommender System is a system that offers personalized recommendations to on-line customers. *"Personalized recommendations"* says Brent Smith, Amazon's director of personalization, *"are at the heart of why online shopping offers so much promise."*

A Recommender system will potentially suggest to the end user to watch or not a movie, to buy or not an item, to listen or not to a song and so forth. In this sense, an accurate Recommender System will ideally be able to act on behalf of the user. To achieve its goal, it must gain knowledge of the user's value system and decision policy.

More formally, a Recommender System can be defined as follow:

Let U be a set of candidate users to receive recommendations and let I be the set of items to be recommended. The goal of a Recommender System would be to maximize the function f that measures usefulness of item i to user u for every user and item:

$$\forall u \in U, i_u = \arg\max f(u,i), \;\; i \in I \qquad \textbf{2.4.1-1}$$

In other words, a successful Recommender System will be able to recommend the most useful item to every user.

In **Figure 2.2-1** the general idea of a Recommender System is shown in a cartoon sketch. The input of any Recommender System would be some kind of preferential information of users and items. This preferential information is processed together with other kind of information that may be stored in the form of user or item profiles and according to the recommendation algorithm, the output of a Recommender System would be the matching of items to users; either single items to single users or any combination of group of items and user groups.

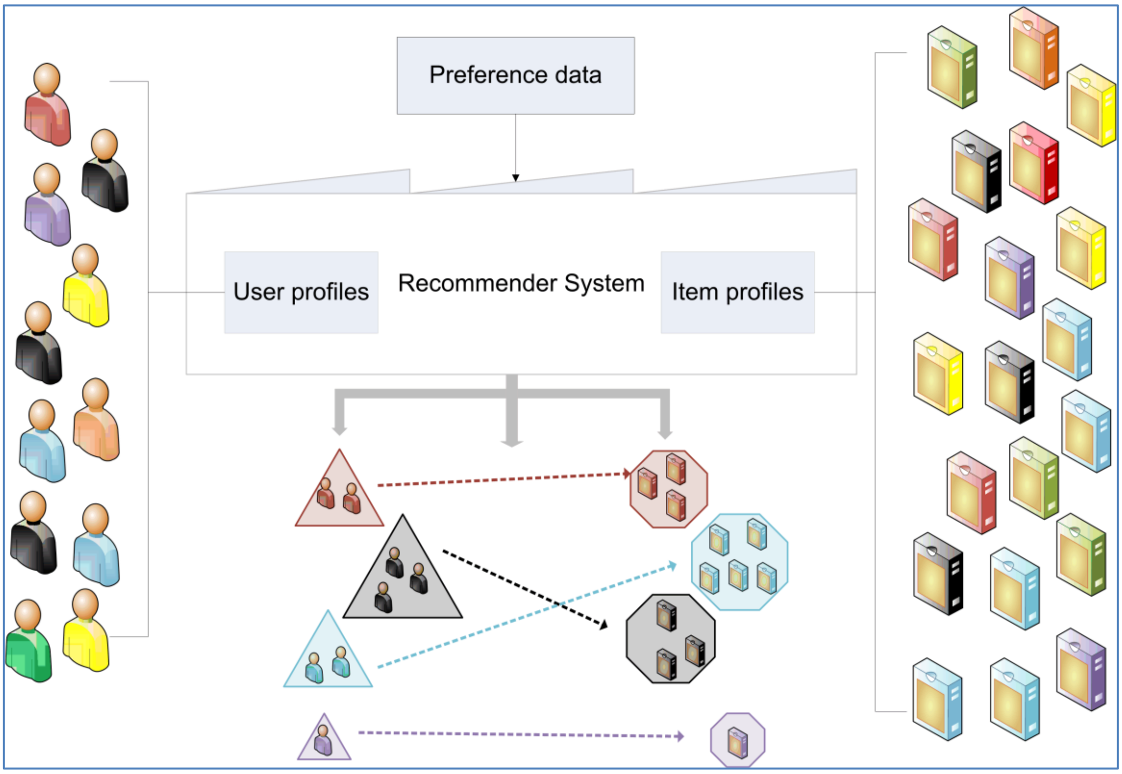


Figure 2.2-1: *A schematically generic representation of a Recommender System*

2.3 Most popular Recommender Systems

Many Recommender Systems have attracted much attention over the years. Popularity is considered a critical aspect for such systems since, apart from its contribution to effectiveness, popularity, helps Recommender Systems to improve their accuracy to a great extent, by providing profusion of feedback. Some of the most popular Recommender Systems besides Grouplens (J. A. Konstan, B. N. Miller, et al. 1997), Movielens (B. J. Dahlen, J. A. Konstan, et al. 1998), Phoaks (L. Terveen, W. Hill, et al. 1997), Referral Web (H. Kautz, B. Selman, et al. 1997) and Fab (M. Balabanović and Y. Shoham 1997), that have already been mentioned in 2.2, are concisely described below. All of them are commercial Recommender Systems that gradually attracted many web users. To this end, there are no analytical details behind the recommendation algorithms that these systems incorporate.

Last.fm (http://www.last.fm/) is a free Internet radio station that uses a recommender system based on tracking what users listen to and makes suggestions based on the users' tastes. It was founded in 2002 by Felix Miller, Martin Stiksel, Saulyus Chyamolonskas, Michael Breidenbruecker and Thomas Willomitzer. By using a music recommender system called "Audioscrobbler", Last.fm builds a detailed profile of each user's musical taste by recording details of all the songs the user listens to, either on the streamed radio stations, or the user's computer or many portable music devices. Recommendations are calculated using a collaborative filtering algorithm, so users can browse and hear previews of a list of artists not listed on their own profile, but which appear on those of others with similar musical tastes.

StumbleUpon (http://www.stumbleupon.com/) is a Recommender System for web surfers, which combines collaborative human opinions with machine learning of personal preference, to create virtual communities of like-minded web surfers. Rating websites updates a personal profile (weblog) and generates peer networks of web surfers linked by common interest. These social networks coordinate the distribution of web content, such that users 'stumble upon' pages explicitly recommended by friends and peers. By October 2009 it claims 8,399,700 users.

Strands (http://www.strands.com/) social Recommender System builds models of how people interact with information consisting of hierarchically structured links between items. When a user requests recommendations, the system consults the models to find a set of possible recommendations. The recommender then filters this initial set of possible recommendations using knowledge of the user's preferences, previous recommendations, and the context of the recommendation request to produce a final set of personalized recommendations. The recommendation system inherently is content and platform agnostic, content awareness is captured in the models for the relationships between items which may be provided explicitly or automatically learned over time. Real-time recommendations are available for new users with no training and increasingly personalized as the system accumulates knowledge about the user. Similarly, new items inserted into the models are immediately recommendable based on approximate relationships with previously existing items and then increasingly recommended based on their relationships with other items learned from user behavior. Strands, recently has extended its services from music discovery to videos and financial products recommendations offered by moneyStrands.

Netflix (http://www.netflix.com/) is a DVD movie rental site. Netflix's Recommender System Cinematch, will recommend a movie or movies based on member reviews, critic reviews, popular rental lists and how active user rates movies. When a user signs up for this service he/ she must submit a list of DVDs that would like to rent from an extensive list of movies. Every user can receive from one to four DVDs at a time; when user returns a DVD to Netflix, they send to the user another from user's list. Several rental plans are available with options as to the number of DVDs that can be ordered per month. On October 2, 2006 Netflix announced the Netflix Prize open competition for the best algorithm to predict user ratings for films, based on previous ratings. The competition was open to anyone and the grand prize of $1,000,000 was reserved for the entry which would improve Netflix's own algorithm for predicting ratings by 10%. On September 21, 2009 they awarded the $1,000,000 Grand Prize to team "BellKor's Pragmatic Chaos". This team improved to two of the most popular so far approaches to Collaborative Filtering. First, they suggested a new neighborhood based model, which unlike previous neighborhood methods, is based on formally optimizing a global cost function. Second, they introduced extensions to Singular Value Decomposition-based latent factor models that allow improved accuracy by integrating implicit feedback into the model.

Pandora™ (http://www.pandora.com/) is a music recommendation system, emerged from the Music Genome Project™, a comprehensive analysis of music in which a team of musician analysts listen to and studies each song and note nearly 400 attributes of each. Pandora makes this information available to the public. Users enter their favorite songs and artists into Pandora, and then Pandora recommends music similar to their taste. Then users can listen to the new music through the site.

Letizia & Let's Browse. Let's browse and its predecessor, Letizia (H. Lieberman 1995), are web agents that assist a user during his/her browsing experience. By monitoring a user's behavior, or browsing time on a web page, Letizia system learns the user's interests and provides recommendation. Let's Browse (H. Lieberman, N. W. V. Dyke, *et al.* 1998), an improved version of Letizia, provides recommendation by using group's profiles instead of using a single profile. If multiple users are reading the same page at the same time, Let's Browse can determine which users are in the area of monitor, and use their profiles to provide recommendation sites for entire group.

Firefly system is based on similarities of users to provide recommendation. At the beginning, this system was used for music and movies recommendation. Later, it was extended to other media recommendation, such as newsgroup, books, and web pages. The system used users' profiles as input, and used constrained Pearson algorithm to make the best predictions between users. The basic idea of the algorithm is: a) the system maintains a user profile, which includes "like or dislike" of specific items, b) the system compares the similarities of users and decides which kind of users that the user belongs, and c) according to the similar user's profile and gives a recommendation. Firefly technology was used by quite a number of well known businesses, including the recommendation engine for barnesandnoble.com, ZDnet, launch.com (later purchased by Yahoo) and MyYahoo. In April 1998, Microsoft Inc. purchased Firefly.

Amazon (http://www.amazon.com/), uses a Recommender System that is based on item-to-item collaborative filtering to overcome scalability issues that are inevitably apparent in user based collaborative filtering systems. By clicking on the "Personalized Recommendations" link customers are driven to an area where they can filter their recommendations by product line and subject area, rate the recommended products, rate their previous purchases, and see why items are recommended. Rather than matching the user to similar customers, item-to-item collaborative filtering matches each of the user's purchased and rated items to similar items, then combines those similar items into a recommendation list. To determine the most-similar match for a given item, the algorithm builds a similar-items table by finding items that customers tend to purchase together.

Google (http://www.google.com/), the most successful internet company of this era, is using recommendation technologies to improve its core search product. There are two ways that Google does this; either by customizing user's search results based on the location and/or recent search activity, or via user's web history, when the user has signed in to his/ her Google Account.

iLike (http://www.ilike.com/) was launched on October 2006 by Ali Partovi with his brother Hadi. Described as a "social-music discovery service", iLike has been incorporated in leading social networks such as Facebook, MySpace and Hi5, making it accessible to more than 30 million registered users. It enables users to recommend music to other users and informs them of upcoming concerts by artists, whose music they are listening to on iTunes. Ticketmaster paid $13.3M for a 25% stake in 2006.

iTunes Genius was launched on September 2008 by Apple Inc. Genius allows iTunes Store customers to create playlists based on the songs' similarity to a chosen track from their collection. Additionally, its 'Sidebar' uses technology that automatically recommends other tracks from the iTunes store itself. The more users employ the system for their own playlists, the more the technology is able to recognize which songs might appeal to them.

Tivo (http://www.tivo.com/), the Digital Video Recorder service, finds and records user's favorite shows, every time they're on, so they can be watched at anytime. Tivo Suggestions is a service offered by Tivo where users can rate programs from three "thumbs up" to three "thumbs down." TiVo user ratings are combined to create a recommendation, based on what TiVo users with similar viewing habits watch.

2.4 Basic methodological approaches in Recommender Systems

The underlying techniques used in today's recommendation systems fall into two major categories, **a)** *content-based filtering* and **b)** *collaborative filtering methods*. However, some works classify existing Recommender Systems into additional categories based for example on their application domain (K. N. Rao and V.G.Talwar 2008), or on a Human-Computer Interaction (HCI) perspective (L. Terveen and W. Hill 2001). Schafer, Konstan, and Riedl (J. B. Schafer, J. Konstan, et al. 1999) survey recommender systems in e-commerce, based on interface technology used to create the recommendations, and the inputs they need from customers, while (S. Perugini, M. A. Gonzalves, et al. 2004) review recommender systems from their social element perspective, meaning the way they model users and resulting connections they achieve or identify. Montaner et al, (M. Montaner, B. López, et al. 2003) have classified recommender systems by following two different approaches. The first, the spatial approach, produces a classification of systems according to the application domain, while the second, the functional approach produces a classification based on the different task-achievement techniques used in the system.

In this thesis, an aggregated presentation of the several categories met throughout literature, intends to provide a more comprehensive and integrated picture of the various types of Recommender Systems. Thus, seven different categories are described, with emphasis on the two major and globally accepted types of such systems, the content-based filtering (CBF) and the collaborative filtering (CF).

2.4.1 Content-based filtering recommenders

The content-based filtering uses actual domain content features of items to generate recommendations. A content-based filtering system selects items based on the correlation between the content of the items and the user's preferences information. In these systems, the process of recommendation first starts by gathering content data about the items. For example, title, author, descriptors, etc. for the books or the director, cast, etc. for the movies, are some of the common content information. Most of these systems use feature extraction techniques and information indexing to extract the content data.

Content-based recommendation systems may be used in a variety of domains ranging from recommending web pages, news articles, restaurants, movies, and items for sale.

Depending on the domain, data that describe items are met either in some kind of structured form, i.e. a book can be filed in a database with several accompanying attributes such as author, length, genre, editor, or in unstructured forms. The latter, requires techniques from natural language processing (such as stemming, stop words removal, or polysemy processing) to extract appropriate attributes that would potentially characterize a specific item. Data can be also met in some kind of semi structured form that is a combination of both attributed items and free text.

Content-based recommendation systems recommend an item to a user based upon a description of the item and a profile of the user's interests. While a user profile may be entered by the user, it is commonly learned from feedback the user provides on items. A variety of learning algorithms have been adapted to learning user profiles, and the choice of learning algorithm depends upon the representation of content (M. J. Pazzani and D. Billsus 2007).

The most popular way to associate a value to every item is the term-frequency, inverse term frequency or simply the *tf-idf* algorithm. The *tf–idf* weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection. The *tf-idf* weighting scheme assigns to term *t* a weight in document *d* that is highest when *t* occurs many times within a small number of documents, lower when the term occurs fewer times in a document, or occurs in many documents and lowest when the term occurs in virtually all documents. The term count in the given document is simply the number of times a given term appears in that document. It is usually normalized to prevent a bias towards longer documents. Let n_{ij} be the number of occurrences of the term t_i in document d_j. Then, tf_{ij} will be given by equation 2.4.1-1:

$$tf_{ij} = \frac{n_{ij}}{\sum_{K} n_{kj}} \qquad \textbf{2.4.1-1}$$

The denominator corresponds to the total number of occurrences of all terms in the document d_j.

Denoting as usual the total number of documents in a collection by N, the inverse document frequency of a term t_{ij} in document d_j ,idf_{ij} is defined as follows 2.4.1-2:

$$idf_{ij} = \log \frac{N}{df_j} \qquad \textbf{2.4.1-2}$$

df_i is defined to be the number of documents in the collection that contain the term t_i. Then, *tf-idf* weight of a term *i* in document *j*, is simply given by: $tf_{ij} \times idf_{ij}$.

Hence, the content of document d_j is defined as in 2.4.1-3:

$$content_j = f(w_{1j}, w_{2j}, ..., w_{Kj}) \qquad \textbf{2.4.1-3}$$

On the other hand, not only items must be represented in a content based approach but also users. Users are is most cases represented by the so called user profiles. A user profile may include information regarding user preferences, user past behavior or both. A record of user history, that may be any kind of past actions, can be used as a filter to produce recommendation or as training data for a machine learning algorithm that creates a user model Several approaches exist to create a user model that are analytically discussed in chapter 3. Numerous techniques for user modeling have been applied to content based recommender systems (M. J. Pazzani and D. Billsus 2007), mainly adopted from Machine Learning, and range from simple classifiers that classify items into “liked” or “disliked” based on some explicit or implicit preference information given by the user, to more sophisticated algorithms that learn a function that models each user’s interests, like decision tree learners such as ID3 (R. J. Quinlan 1986), rule induction algorithms (W. W. Cohen 1995), nearest neighbor algorithms (D. Billsus, M. J. Pazzani, et al. 2000), linear classifiers (T. Zhang and V. S. Iyengar 2002), naïve Bayesian classifiers (M. Pazzani, J. Muramatsu, et al. 1996) and other.

Content-based filtering systems have several advantages:

- They don't require data on other users and are away from new user cold-start and sparsity problems, since they don't require any user to user similarity calculations.
- These systems are capable of recommending items to users with unique tastes.
- They can provide explanations of recommended items by explicitly listing content features or descriptions that caused an item to be recommended.
- They do not suffer from first-rater problem, i.e., they are capable of recommending new and unpopular items to each and every user.

However, these methods also suffer from several limitations that are discussed in 2.5.1.

2.4.2 Collaborative filtering recommenders

Collaborative filtering (or social filtering) recommenders use people's opinions of items in the domain to generate recommendations assuming the like-minded people tend to have similar choices. The collaborative filtering (CF) can be seen as the computer automated "word of mouth" process. Computers and the Web allow billions of users to express their opinion making this approach even more reliable. In a typical collaborative filtering system the user preference data consist of the so called *users* × *items* matrix, where usually the rows correspond to users and columns to items and each entry of this matrix denotes the preference of user *u* for the item *i*. One major and common deficiency of this input that limits significantly the performance of any collaborative filtering algorithm used to predict values for empty cells of this matrix, is its low density, named as sparseness. A detailed description however of the limitations of this approach is provided in 2.5.2.

Several collaborative filtering algorithms exist in the literature (J. B. Schafer, D. Frankowski, et al. 2007) and a plethora of Recommender Systems that exploit them. Usually, collaborative filtering systems are classified into two categories, the *memory-based algorithms* that make rating predictions based on the entire collection of previously rated items by the users and *the model-based algorithms* that use the collection of ratings to learn a model, which is then used to make rating predictions (J. S. Breese, D. Heckerman, et al. 1998). Moreover, Schafer et al (J. B. Schafer, D. Frankowski, et al. 2007), classifies collaborative filtering algorithms into non-probabilistic algorithms if they are based on an underlying probabilistic model and non probabilistic otherwise.

According to Koren (Y. Koren 2008), two are the most successful approaches to collaborative filtering, *latent factor models*, which directly profile both users and products, and *neighborhood models*, which analyze similarities between products or users.

Neighborhood methods are based on computing the relationships between items or, alternatively, between users. An item-oriented approach evaluates the preference of a user to an item based on ratings of similar items by the same user. In a sense, these methods transform users to the item space by viewing them as baskets of rated items. This way, there is no longer need to compare users to items, but rather directly relate items to items.

Latent factor models, such as Singular Value Decomposition (SVD) models, comprise an alternative approach by transforming both items and users to the same latent factor space, thus making them directly comparable. The latent space tries to explain ratings by characterizing both products and users on factors automatically inferred from user feedback. For example, when the products are movies, factors might measure obvious dimensions such as comedy vs. drama, amount of action, or orientation to children; less well defined dimensions such as depth of character development or “quirkiness”; or completely uninterpretable dimensions.

Neighborhood models are most effective at detecting much localized relationships. They rely on a few significant neighborhood relations, often ignoring the vast majority of ratings by a user. Consequently, these methods are unable to capture the totality of weak signals encompassed in all of a user's ratings. Latent factor models are generally effective at estimating overall structure that relates simultaneously to most or all items. However, these models are poor at detecting strong associations among a small set of closely related items, precisely where neighborhood models do best.

Based on whether the similarity is calculated among users or among items, collaborative filtering approaches are divided into *user based* and *item based* respectively. The notion of similarity has been defined in many different ways, however the two most commonly used similarity measures are **a)** the cosine similarity and **b)**, the Pearson correlation coefficient that are given in equations 2.4.2-1 and2.4.2-2, respectively.

In a simple cosine based similarity equation, the value of similarity for two users depends on a combination of the ratings *R(u,i)* that both user *u* and user *u'* have given in the past for every item *i* of the entire set of common items *I*.

$$sim(u,u') = \frac{\sum_{i \in I(u,u')} R(u,i) \cdot R(u',i)}{\sqrt{\sum_{i \in I(u,u')} R(u,i)^2} \cdot \sqrt{\sum_{i \in I(u,u')} R(u',i)^2}} \qquad \textbf{2.4.2-1}$$

Common items set is referred to the set of items that both users *u* and *u'* have rated and is denoted here as *I(u,u')*.

Pearson correlation calculates similarity *sim(u,u')* by equation 2.4.2-2:

$$sim(u,u') = \frac{\sum_{i \in I(u,u')} \left(R(u,i) - \overline{R}(u)\right) \cdot \left(R(u',i) - \overline{R}(u')\right)}{\sqrt{\sum_{i \in I(u,u')} \left(R(u,i) - \overline{R}(u)\right)^2} \cdot \sqrt{\sum_{i \in I(u,u')} \left(R(u',i) - \overline{R}(u')\right)^2}} \qquad \textbf{2.4.2-2}$$

This coefficient ranges from 1 for users with perfect agreement (positive correlation) to -1 for perfect disagreement users (negative correlation).

Calculating similarity is just the first step of any collaborative filtering approach. Both, user based and item based systems must follow an algorithm that defines which users or items will be used in predicting a rating. The most obvious and straightforward approach is to weight similarities. Two common approaches of weighting similarities, either user based or item based are a) the *weighted sum approach* and b) the *adjusted weighted sum approach* (G.

Adomavicius and Y. O. Kwon 2007), given in equations 2.4.2-3, 2.4.2-4 and 2.4.2-5 respectively.

$$R(u,i) = z \sum_{u' \in N(u)} sim(u,u') \cdot R(u',i) \qquad \textbf{2.4.2-3}$$

$$R(u,i) = \overline{R(u)} + z \sum_{u' \in N(u)} sim(u,u') \cdot \; R(u',i) - \overline{R(u')} \qquad \textbf{2.4.2-4}$$

$$z = 1 / \sum_{u' \in N(u)} | sim(u,u') | \qquad \textbf{2.4.2-5}$$

Factor z serves as normalization multiplier and is defined for the total neighbors $N(u)$ of user u.

Other approaches, more sophisticated, have been developed under the collaborative filtering philosophy, trying to achieve better prediction accuracy. For instance, association mining techniques build models based on commonly occurring patterns in the *users* × *items* matrix (W. Lin and S. A. Alvarez 2002). These rules are created based on users past behavior, on observation for example that users who rated item 1 highly, often rate item 2 highly. Associations can be employed either between users or between items in making recommendations. Moreover, numerous dimensionality reduction algorithms like Factor Analysis (J. Canny 2002), Singular Value Decomposition (B. Sarwar, G. Karypis, et al. 2002), Principal Component Analysis (D. Kim and B.-J. Yum 2005) e.t.c., have been applied to reduce the complexity of the *users* × *items* matrix by identifying important latent parameters hiden in its vastness.

Probabilistic approaches have been also proved useful in collaborative filtering systems. In general such approaches try to compute the probability P that a user u will give a particular rating to item i, given that user's ratings of the previously rated items. Simple probabilistic approaches like Bayesian networks (J. S. Breese, D. Heckerman, et al. 1998) are often used in such approaches. More complex probabilistic models like Markov Decision Processes (MDP's) (G. Shani, D. Heckerman, et al. 2005) are also met in Collaborative Filtering literature. In the last approach, the recommendation process is considered as a sequential decision problem and MDP's, a well-known stochastic technique for modeling sequential decisions is applied to generate recommendations. Recently, Campos *et al.* (L. M. d. Campos, J. M. Fernández-Luna*, et al.* 2008), combine Bayesian networks and Fuzzy Set Theory to automatically suggest and rank a list of new items to a user based on the past voting patterns of other users with similar tastes.

Collaborative filtering systems encompass the following advantages:

- They do not need a representation of items in terms of features i.e. genre and actor of the movie, title and author of the book, but they are based only on the judgment of participating users community. Hence, collaborative filtering can be applied to virtually any kind of item, i.e., papers, news, websites, movies, songs, books, jokes, etc.
- Scalability of the items database can be large since the technique does not require any human involvement for tagging descriptions or features.
- They can cope with cross-genre recommendations such as making predictions of entirely different items to a user who has never rated such items in the past.
- They do not require domain-knowledge for linking the features to the items.
- There is high potential of improved recommendations over time.

2.4.3 Knowledge-based recommenders

Knowledge-based recommenders use rules, patterns, or connections between items to generate recommendations (e.g. when you are buying a lamp, it suggests that you also buy some light bulbs). In other words, such systems use the functional knowledge to generate recommendations, i.e. knowledge about how a particular item meets a particular user need, and can reason about the relationship between a need and a product. Knowledge based recommenders rely either on explicit domain knowledge about the items or on knowledge about the users.

One major category of knowledge based recommenders is those that rely on Case Based Reasoning (CBR), often attributed as Case Based Recommenders (CBR). In such systems items or products are represented as cases and recommendations are generated by retrieving those cases that are most similar to a user's query or profile (B. Smyth 2007). A simple form of a CBR is shown in **Figure 2.4-1**. Such a system will retrieve and rank product suggestions by comparing the user's target query to the descriptions of products stored in its case base using similarity knowledge to identify products that are close matches to the target query.

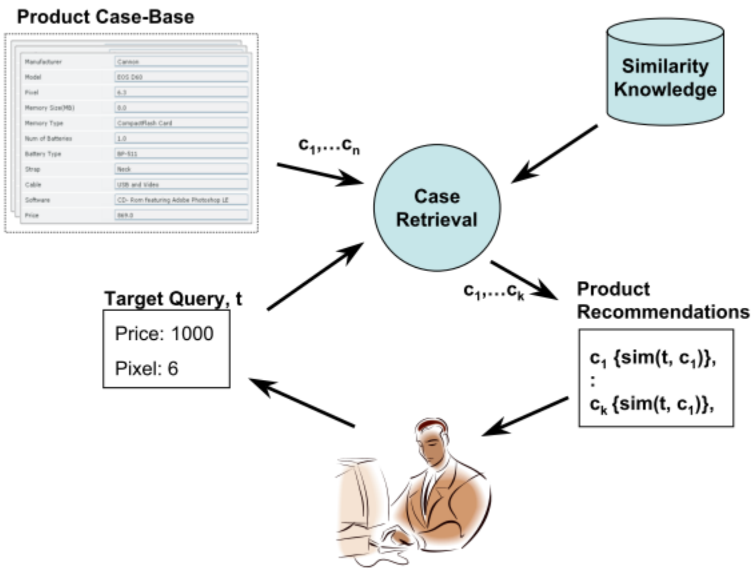


Figure 2.4-1: *A simple form a case-based recommendation system*

The product representation, as well as the way in which product similarity is assessed, distinguish content based recommenders from other recommender systems. One of the earliest content based recommender system is Entrée (R. D. Burke, K. J. Hammond, et al. 1997), which uses a knowledge base and case-based reasoning to recommend restaurant data. It has an extensive and well constructed database, supports second level of recommendation, overcomes the cold start problem, yet it is static, due to the absence of user evaluation ability.

Another type of knowledge based recommender system are those that incorporate ontologies (S. E. Middleton, D. D. Roure, et al. 2004). According to the definition by Tom Gruber, "*An ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members). The definitions of the* *representational primitives include information about their meaning and constraints on their logically consistent application"*. For example, Middleton *et al* (S. E. Middleton, N. R. Shadbolt, et al. 2004) presented Quickstep and Foxtrot. Quickstep is a recommender system for on-line research papers that uses ontological inference to improve profiling accuracy and integrates an external ontology for profile bootstrapping, while Foxtrot is a searchable database and recommender system for a computer science department and enhances the Quickstep system by employing the idea of visualizing user profiles to acquire direct profile feedback.

2.4.4 Demographic filtering recommenders

Demographic filtering recommenders are systems that use demographic information such as age, gender, education, etc. of people for identifying types of users like a certain object and makes recommendations. Demographic filtering (DF) shares the view expressed by collaborative filtering, in that similar users are expected to share the same interests. However, this approach tries to tackle the recommendation problem from a different, somewhat more general perspective. Instead of using the ratings provided by users to compile profiles, users are asked to provide demographic information such as their age, interest in sports, favorite TV programs, and purchasing history, among others. They are then compared to precompiled clusters of the general population, indexed by the same characteristics. Sets of resources available for recommendation are matched a priori with such clusters. Once the most similar cluster has been obtained, the resources associated with it are recommended. The idea of Demographic filtering was first introduced by (B. Krulwich 1997). Krulwich created 62 demographic clusters based on the surveys of more than 40 000 people as well as U.S. census data, magazine subscriptions, and catalog purchases. Using this dataset to train against, they were able to report encouraging results by matching new users to clusters, and then recommending products explicitly associated with members of that cluster.

Demographic filtering mainly suffers from overgeneralization of user interests and is typically used as one of several components in hybrid recommender systems.

2.4.5 Conversational recommender systems

Conversational recommender systems or feedback based recommenders guide users through a product space, alternatively making concrete product suggestions and eliciting the user's feedback. In such systems users are involved into a recurrent procedure and during each cycle of a recommendation session a user is presented with a new recommendation and is offered an opportunity to provide feedback on this suggestion. On the basis of this feedback, the recommender system will revise its evolving model of the user's current needs in order to make further recommendations. Reilly *et al.* (J. Reilly, K. McCarthy, et al. 2005), propose an incremental critiquing strategy and model the user based on the set of critiques that the user has applied in the past.

Viappiani *et al.*(P. Viappiani, P. Pu, et al. 2007), consider a mixed initiative tool for preference based search, in which preferences are stated as critiques to shown examples (user-motivated critiques) and suggestions adapted to users' reactions are presented, to elicit preference information. The interaction cycle of a conversational recommender system with adaptive suggestions that they propose, is shown in **Figure 2.4-2**.

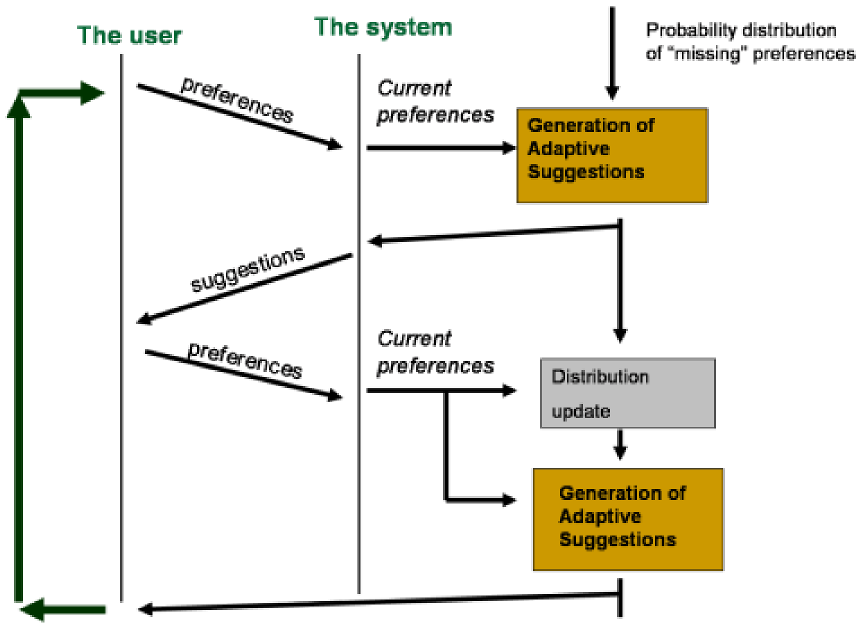


Figure 2.4-2: *A conversational recommender system with adaptive suggestions*

Pear Pu and Boi Faltings (P. Pu and B. Faltings 2000) introduced and developed a travel planning system, called SmartClient, where new techniques for better adapting interaction of users with an electronic catalog system to actual buying behavior was explained. Their model was based on a conversation that supported the buyer in formulating his or her needs and in deciding which criteria to apply in selecting a product to buy. Later, further work in the technology of example critiquing agents confirmed that example critiquing significantly reduces users' task time and error rate while increasing decision accuracy (P. Pu, L. Chen*, et al.* 2008). Li Chen in her PhD thesis (L. Chen 2008) describes two primary technologies: one is called example critiquing agents aimed to stimulate users to conduct trade off navigation and freely specify feedback criteria to example products; another termed as preference-based organization interfaces designed to take two roles: explaining to users why and how the recommendations are computed and displayed, and suggesting critique suggestions to guide users to under-stand existing trade off potentials and to make concrete decision navigations from the top candidate for better choices.

Ardissono et al (L. Ardissono, A. Goy*, et al.* 2003), presented INTRIGUE (INteractive TouRist Information GUidE), a tourist personalized information recommendation service that presents information about a specific geographical area around Torino city, on desktop and handset devices. The recommendation activity is based on a declarative representation of the knowledge about tourist attractions and on the application of fuzzy evaluation functions for ranking the items. INTRIGUE explains the recommendations with respect to the preferences of the tourist group members and offers an interactive agenda that helps the user to schedule the tour.

2.4.6 Hybrid recommenders

Hybrid recommenders combine two or more categories of Recommender Systems in order to overcome certain limitation of the individual categories. For example, Karacapilidis and Hatzieleftheriou (N. Karacapilidis and L. Hatzieleftheriou 2003) combine both, knowledge based and collaborative filtering approaches, into a hybrid framework that exploits the concept of fuzzy similarity measures in order to make accurate recommendations.

According to Burke (R. Burke 2005), there are mainly seven ways by which recommender systems are combined into hybrid frameworks:

- Mixed, where different recommenders are combined together and their results are presented together either in a combined presentation or in separate lists.
- Weighted, wherein scores from the recommenders are combined using weights to derive a single score.
- Switching, where the system uses some decision criteria to choose a recommender based on the context and uses the results from only the chosen source.
- Cascade, when one recommender refines the recommendations produced by another.
- Feature Combination, where data from different source types are combined together and treated using one recommendation algorithm.
- Feature Augmentation, when the output from one technique is used as an input feature to another.
- Meta-level in which one recommender produces a model, which is then used as input for the second recommender.

Albadvia, A. and M. Shahbazi (A. Albadvia and M. Shahbazi 2009) introduce a technique of recommendation in the context of online retail store which extracts user preferences in each product category separately and provides more personalized recommendations through employing product taxonomy, attributes of product categories, web usage mining and combination of collaborative filtering (CF) and content-based filtering (CBF). User profile is created by implicit ratings of user to product attributes. According to sequential combination model in hybrid recommender systems, customer profile is created by CBF approach and consequently, CF is applied to improve recommendation accuracy. Web usage mining is employed to analyze customers' shopping behaviors on the web and collects their implicit ratings on the attributes.

Numerous approaches appear in the literature every day, most of them designed under hybrid frameworks. With the growth of Recommender Systems technology and the almost saturated development of systems with individual technologies, hybrid systems seem to spread exponentially.

2.4.7 Multi-criteria Recommender Systems

As stated in (S. M. McNee, J. Riedl, *et al.* 2006), Recommender Systems need a deeper understanding of users and their information seeking tasks. They propose the Human-Recommender Interaction (HRI) framework to incorporate knowledge for the user into the Recommender System. Moreover, the article published in the Wall Street Journal by (J. Zaslow 2002) underlies the pretence for stirring the Recommender Systems researchers towards a more user oriented perspective. This means that Recommender Systems must understand not only what users think of items but also why they think so.

Multi-criteria Recommender Systems are defined as Recommender Systems that incorporate preference information upon multiple criteria. Instead of developing recommendation techniques based on a single criterion values, the overall preference of user u for the item i, these systems try to predict a rating for unexplored items of u by exploiting preference information on multiple criteria that affect this overall preference value. So far, various ad hoc attempts have been made towards this direction and no unified framework exists for the researchers to follow (S. Tsafarakis, K. Lakiotaki, *et al.*). Multi-criteria Recommender Systems are still considered a premature, however highly promising field of research.

The majority of existing multi-criteria collaborative filtering recommender systems adopt the Multi-Attribute Utility Theory (MAUT) (R. Keeney and H. Raiffa 1993) and engage some linear additive value function for aggregating user preferences of multiple criteria into the total user preference.

A first demonstration of exploiting methodologies originated from Decision Theory to model user judging policy and deploy that information in the recommendation process can be found in (N. Matsatsinis, K. Lakiotaki, *et al.* 2007). There, the authors deal with the problem of recommending academic papers to the research community by introducing a specific methodology that was demonstrated as a pioneering example for the incorporation of Multiple

Criteria Decision Analysis (MCDA) techniques to Recommender Systems. With the proposed system a researcher could choose among the vast amount of literature available, the best publications concerning his/her preferences and focus explicitly on them. The discussed system offered the user/decision maker the opportunity to express his/her dynamic intentions/preferences and in that sense it was utterly user-oriented.

Recently, Zhang *et al.* (Y. Zhang, Y. Zhuang, et al. 2009), studied the statistical machine learning methods in the context of multi-criteria ratings, by applying the Probabilistic Latent Semantic Analysis (PLSA) from a multi criteria perspective. They approached the recommendation problem by considering that overall rating is the result of multiple criteria ratings and they exploit linear Gaussian regression to model the relation between overall rating and criteria ratings. They actually replaced the univariate Gaussian distribution for single rating PLSA with multivariate distribution in two different ways. In their first approach, they considered the criteria and overall preference information as a single preference vector, while in their second approach they exploited the linear Gaussian regression model, assuming that multi criteria ratings are dependent and overall rating is the linear Gaussian regression of these individual ratings. Both their proposed methods were proved to perform better than the tested single-criterion and multi-criteria methods (an item based Pearson correlation coefficient approach and a linear regression coefficient approach) in terms of prediction and ranking accuracy.

Karacapilidis and Hatzieleftheriou (N. Karacapilidis and L. Hatzieleftheriou 2003) proposed a hybrid framework that incorporates a multi-criteria approach to manage and evaluate recommendations and demonstrate their approach via the City Guide recommender tool.

Hsin-Hsien Lee and Wei-Guang Teng (W.-G. Teng and H.-H. Lee 2007) used data query techniques to solve the multi-criteria recommendation problem. They formulated the problem of finding an optimal solution to a decision with multiple conflicting criteria by exploiting the notion of skyline queries from the database field. Skyline queries are based on the dominance principle as referred in the theory of multiple criteria analysis. According to this principle, given a d-dimensional dataset, a point p is said to dominate another point q if it is better than or equal to q in all dimensions and better than q in at least one. A skyline then is a subset of points that are not dominated by any others. Skyline queries are defined as those queries which return skyline points.

Sahoo *et al* (N. Sahoo, R. Krishnan, et al.), presented an approach to integrate multi-criteria rating into a collaborative filtering algorithm. They observed that the overall rating, an indicator of rater's general preference, is the highest correlation inducing variable. By controlling the overall rating they observed that the remaining criteria ratings become more independent. Subsequently they incorporated this conditional independence into a mixture model describing rating generation process and exploited the EM algorithm to estimate parameters of the model. They found that by using a small training data set they achieved better prediction and retrieval; however when they used a sufficient training data set the gain was relatively small.

Chapphannarungsri and Maneeroj (K. Chapphannarungsri and S. Maneeroj 2009) identified neighbors of a target user from three vectors; **a)** User Preference Vector (UPV), **b)** Selection on Movie Features Vector (SMV) and **c)** Multidimensional Vector (MDV). The UPV represents a user's opinion on specific features and it is based on the Movie Feature Vector (MFV), which contains 24 elements (18 elements of movie genre feature, 3 elements of year feature and 3 elements of award feature. UPV for every user is constructed by calculating the direct sum of the transformed MFV of rated movies and divided by the number of rated movies by the specific user. SMV vector contains 7 elements of movie selection features which are title, genre, release, period, actor, actress, director and award. Finally, MDV concerns the multidimensional nature of the Recommendation problem and is constructed by Multiple Linear Regression. The researchers demonstrated their methodology by the MoviePlanet System, a multi-rating, multidimensional movie Recommender System. This is a very recent work of Recommender System that employs multi rating information by considering the weights that affect the overall preference value, together with contextual information.

Le Roux et al (F. L. Roux, E. Ranjeet, et al.), developed a course recommender system capable of helping prospective students to choose relevant post graduate courses by a multiple criteria decision making method. The recommender system uses simplistic calculations to find out the least distance between the course preferences set by the user and the course values defined in the database.

Choi et al (S. H. Choi, S. Kang, et al. 2006), proposed a recommendation system which enables bidirectional communication between the user and the system using a utility range-based product recommendation algorithm. They calculated similarity by exploiting a Multi-Attribute Decision Making (MADM) to find the utility values of products in same product class of the companies and asked the user to provide the information about the criteria weights in format of incomplete information in terms of relationships among the weights and ranges of weight values.

Perny and Zucker (P. Perny and J. D. Zucker 2001) constructed a preference-based filtering relying on the integration of content-based and collaborative filtering principles. They also offered the ability to explain and justify recommendations. To achieve their goal they integrated preference modeling and machine learning and they demonstrated their approach by presenting the “film-conseil” system for movie recommendation tasks on the internet. In their work they introduced the notion of “Collaborative decision support” (CDS) as a new category of decision or search problems where any individual seeks recommendation for his personal choices, the other individuals being only considered as possible advisors. They formulate a fuzzy relational system as the basis of their filtering methods. They compute the similarity of items based on a multi-attribute analysis of items.

Manouselis and Costopoulou (N. Manouselis and C. Costopoulou 2007) identified a set of dimensions that distinguish, describe and categorize multi-criteria recommender systems, based on existing at that time taxonomies and categorizations and integrated these dimensions into an overall framework that was used for the analysis and classification of 37 different multi-criteria recommender systems.

However, most of the 37 systems mentioned in that analysis designate evaluation issues and many were presented in a prototype basis. None of these studies is implemented in real multi-criteria preference user data, raising issues of validity and robustness. Clearly the analysis shows that multi-criteria recommenders systems have been developed sporadically, in an ad hoc manner showing that this research field is still in its infancy.

According to (N. Manouselis and C. Costopoulou 2007), DIVA, was a collaborative filtering movie Recommender System that its authors described as the first attempt to use decision-theoretic techniques in design of a collaborative filtering system. It represented user preferences using pair-wise comparisons among items, rather than numeric ratings. However, DIVA incorporated multi-criteria information by explicitly asking the user to provide pair wise comparisons via an incremental preference elicitation scheme. Recommendations in DIVA are provided via a constraint search where user may specify actors, directors, genres, professional star ratings, countries of production, release years, e.t.c. Although DIVA included multi criteria information into a constraint search option for the user, it didn't use this information to model user preferences.

At this point, a distinction between multi-criteria and multi-dimensional Recommender Systems should be made that is crucial for every researcher in this field. Multi criteria Recommender Systems deal with the problem of modeling user preferences on items and they approach this issue by considering that different, nonetheless related and even conflicting attributes, are hidden behind an overall preference value of an item. Their goal is to reveal the policy under which a user decides to rate an item, by calculating for example criteria weights (trade-offs).

In contrast, multi-dimensional Recommender Systems deal either with considering several dimensions that influence user preferences, such as time, location, weather e.t.c, or several attributes that although affect the overall preference on those items, no individual preference information on those attributes is elicited . For example, Adomavicius et al (G. Adomavicius, R. Sankaranarayanan, et al. 2005) present a multidimensional (MD) approach to recommender systems that can provide recommendations based on additional contextual information. Recently, more and more researchers and practitioners in many disciplines, including e-commerce personalization, information retrieval, ubiquitous and mobile computing, data mining, marketing, and other, identify the importance of contextual information. This is also verified by the fact that a special workshop was dedicated to the issue of incorporating contextual information in Recommender Systems, the Workshop on Context-Aware Recommender Systems (CARS-2009) organized together with the single international conference exclusively devoted to Recommender Systems (RecSys 2009).

Conclusively, it is crucial to shed light to the confusion that exists in the notation of multi-criteria Recommender Systems. For the purposes of this work, multi-criteria Recommenders are defined as Recommender Systems that incorporate multi-criteria preference information. As such, preference evaluation is not solely based on an overall preference value. On the contrary, preference values on the several criteria that affect the total preference need to be known. Nevertheless, the majority of the researchers refer to multi-criteria or multi-attribute Recommenders as long as items can be characterized by multiple attributes, independently on whether users evaluate items on those individual attributes, sometimes also referred as item characteristics.

2.5 Limitations of existing approaches

Recommender algorithms face several problems. In this section, these problems are summarized with the aim of referring back to these problems and discuss how the proposed methodology limits some of these problems and to what extent. The limitations of existing Recommender Systems are presented according to the category that they belong.

2.5.1 Limitations of content – based Recommenders

- The feature extraction and representation can be achieved automatically for machine parsable items such as news or papers. But humans have to manually insert the features for items that are not machine parsable such as movies and songs. The activity of human involvement is highly subjective, expensive, time consuming and erroneous. Moreover, it is impossible to define a right set of features for some sort of items such as jokes for example. In non-textual domains like movies and audio, many content algorithms cannot successfully and reliably analyze item contents. Rich metadata, such as actors, directors, artists, etc., has been improving recommendations in this area, but does not overcome the problem of analyzing non-textual content.

- Content-based filtering techniques have no inherent method for finding something unexpected and useful while searching for something else. The system recommends only more of what the user has already seen and indicated as "liking" the item. Hence, the user is restricted to see items similar to those already rated and these systems suffer from new-user cold start problem. Content-based filtering recommends items similar in content to previous items, and cannot produce recommendations for items that may have different but related content.
- In content-based filtering systems, items are limited to their initial descriptions or features. This limitation makes the content-based techniques dependent on the features that are specified explicitly. Moreover subjective aspects of the content in an item, such as style and quality of writing are hard to analyze. Writing samples can be grammatically analyzed, and thus some level of quality can be achieved. But this is not a semantic analysis; the meaning of the content cannot be easily determined through automatic methods.

2.5.2 Limitations of collaborative filtering based Recommenders

Collaborative algorithms, are generally "domain independent" in that they perform no content analysis of the items in the domain. Rather, they rely on user opinions to generate recommendations. Despite being a successful technique in many domains, collaborative algorithms have their share of shortcomings.

- Cold Start a.k.a. the First-Rater Problem concerns the issue that the system cannot draw any inferences for items about which it has not yet gathered sufficient information. When a new item is added to the database, the item cannot be recommended to any user until the item is either rated by another user(s) or correlated with other similar items of the database. When a collaborative system is first created, there are many items in the system, few users, and no ratings. Without ratings, the system cannot generate recommendations and users see no benefit. Without users, there is no way for new ratings to be entered into the system. When applying these algorithms to a new domain, it is valuable to seek preexisting data that can be used to seed such a database of ratings.

- The New-User Problem. The same notion of cold-start is also applied in users. When a user first enters into a recommender system, the system knows nothing about user's preferences. Consequently, the system is unable to present any personalized recommendations. Before a user can take advantage of a collaborative recommender system, the user must first provide some opinions. This problem is common to other varieties of recommenders as well, but is more severe for collaborative recommenders since these recommenders cannot rely on content or categories to 'ease' a user into the system.
- Data Sparseness Problem: In a large ecommerce site such as Amazon.com, there are millions of products and so customers may rate only a very small portion of those products. In practice even active users only rate a few of the entire set of items and results in a very sparse. Most similarity measures used in collaborative filtering work properly only when there exists an acceptable level of ratings across customers in common. The sparseness of a collaborative filtering matrix is the percentage of empty cells. Because of the presence of empty cells, it difficult to find agreement among individuals, since they may have little overlap and thus collaborative filtering systems may not locate successful neighbors and generate weak recommendations. Different recommender algorithms deal with this problem in various ways. Item-based collaborative filtering uses similarity measures between items. Item-based collaborative filtering reduces the impact of sparsity on the same dataset than algorithms using a user-item ratings matrix. Statistical-based or latent analysis algorithms, such as Naïve Bayes and PLSI, also work in sparse situations, mining all connection data to generate recommendations.
- Critical Mass Problem: For the recommendations to be reliable, the filtering system needs a very large number of people (typically thousands)

to express their preferences about a relatively large number of options (typically dozens). But developing such a database for achieving a critical mass of participants makes collaborative filtering experiments so expensive and time consuming, because users will not be very motivated to express preferences in the beginning stages when the system cannot yet help them.

- Unusual User Problem: In a small or medium community of users, there are individuals whose opinions or tastes are unusual. This means that an individual does not agree or consistently disagrees with any of existing group of people. So, these individuals rarely receive accurate collaborative recommendations even when the critical mass of users is achieved.
- Popularity Bias: Collaborative filtering systems cannot recommend items to someone with unique tastes, but tends to recommend popular items.

2.6 Evaluation of Recommender Systems

In general, Recommender Systems have been evaluated in many, often incomparable, ways. A review on the key decisions in evaluating collaborative filtering recommender systems can be found in (J. L. Herlocker, J. A. Konstan, et al. 2004). In this thesis focus is given on the ways in which prediction quality is measured. To assess prediction quality, three different kinds of metrics are employed.

2.6.1 Statistical Accuracy Metrics

Statistical accuracy metrics measure how close is the numerical value r'_{ui}, which is generated by the Recommender System and represents the expected rating of user u on item i, to the actual numerical rating r_{ui}, as provided by the same user for the same item. The most commonly used statistical accuracy metric is the Mean Absolute Error (MAE). Since Mean Absolute Error measures the deviation of predictions generated by the Recommender System from the true rating values, as they were specified by the user, it is measured only for those items, for which user u has expressed his opinion. Suppose n is the number of items that user *u* has expressed an opinion, then, the MAE_u is formally given by equation 2.6.1-1.

$$MAE_u = \frac{1}{n}\sum_{i=1}^{n}\left|r_{ui} - r'_{ui}\right| \qquad \textbf{2.6.1-1}$$

The average MAE for an entire data set, can be calculated by averaging the Mean Absolute Errors of all users, MAE_u, for $u = 1, 2, \ldots m$, over the total number of available users *m* and can give as an overall estimation of a model's performance.

Another very popular statistical accuracy metric is the Root Mean Squared Error. The difference in RMSE and MAE is that in MAE all the individual differences are weighted equally in the average, while in RMSE since the errors are squared before averaged, relatively high weight is given to large errors. The equation to calculate RMSE is given as:

$$RMSE_u = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(r_{ui} - r'_{ui}\right)^2} \qquad \textbf{2.6.1-2}$$

Alike average MAE, average RMSE offer a global estimation of a model's prediction accuracy.

2.6.2 Classification accuracy metrics

Classification accuracy metrics determine the success of a prediction algorithm in correctly classifying items. In Recommender Systems, a rational classification of items would be as *"highly recommended"* and *"not recommended"*. Items of the first class are very likely to be proposed by the system, while items that belong to the second category will be never shown to the user.

Precision, the number of true positives, is the number of items correctly labeled as belonging to the *"highly recommended"* class, divided by the total number of items labeled as belonging to the same class. Recall is defined as the number of true positives divided by the total number of elements that actually belong to the *"highly recommended"* class. Since there is a trade-off between precision and recall, F-measure, a harmonic mean that equally weights precision and recall is often used.

Receiver Operating Characteristic Analysis (ROC Analysis) is a useful technique for organizing classifiers and visualizing their performance. It is related to cost-benefit analysis in decision making and has been widely used in medicine. Recently it has been introduced in machine learning (T. Fawcett 2003). ROC graphs are two-dimensional graphs in which True Positive rate is plotted versus False Positive. True positive rate is the actual number of positives correctly classified by a model over the total positives. False Positive Rate is the result of the number of negatives incorrectly classified, divided by the total negatives. An ROC graph depicts relative trade-offs between true positives and false positives. The diagonal line of an ROC graph represents the case of randomly guessing a class. Furthermore, the Area Under the Curve (AUC) has been shown to be an accurate evaluation measure and is widely used in applications where ranking is crucial. The maximum value for the AUC is 1.0, a point in the upper left corner or coordinate (0,1), thereby indicating a (theoretically) perfect classifier. The best possible prediction method would yield AUC values vary between 0 and 1 but since an AUC of 0.5 represents the performance of a random classifier, values less that 0.5 indicate no discriminating power of the model.

2.6.3 Rank correlation coefficient

Kendall's tau is a measure of correlation between two ordinal-level variables. In order to calculate Kendall's tau for any sample of n items, there are $[n\ (n-1)/2]$ possible comparisons of points $(x_i,\ y_i)$ and $(x_j,\ y_j)$. Suppose M_C is the number of pairs that are concordant, M_D is the number of discordant pairs and M is the total number of pairs. By concordant pair we mean that for the specific pair of items, both the user and the model ranked them identically. The formula for Kendall tau τ is:

$$\tau = \frac{M_C - M_D}{\sqrt{(M - I_Y) - (M - I_Y)}} \qquad \textbf{2.6.3-1}$$

In 2.6.3-1, I_Y is the number of equivalent pairs regarding ranking order Y (user's ranking order) and $I_{\hat{Y}}$, is the number of equivalent pairs regarding ranking order $\hat{Y}$ (model's ranking order). Kendall's tau varies between -1 and 1 with 1 indicating a total agreement of the orders.

2.7 Open issues in Recommender Systems

As soon as Recommender Systems emerged as a research topic on its own, several researchers and practitioners have contributed to this field. The first official meeting, a predecessor of the today's International conference on Recommender System (RecSys), was held in Bilbao Spain and organized by Strands. Back then, all participants highlighted the need to be more done in order to make Recommender Systems a more integral part of the user experience. The key question of this statement though, is "towards which direction"?

Definitely, in a relatively new and at some point unexplored field of Recommender Systems, there are many components under which these systems can be developed. The undeniable parameter that should be considered no matter which directions researchers are heading to is "user satisfaction". The goal of any successful recommender system is to fulfill user expectations and the Recommender System that achieves that to the highest degree, wins, independently on what technique is based on.

Any subfield of research in Recommender System such as multi-criteria recommenders, contextual recommenders, e.t.c. that has been already mentioned above is considered an open field of study and research. Subsequently, together with all the aforementioned types of Recommenders, new kinds of research directions appear in the literature and prove that the field of Recommender Systems has a great potential to explore. An extensive, but surely not complete, list of open research issues and recent directions identified throughout literature is provided hereupon.

2.7.1 Group recommenders

Humans are social by nature and there are many real-life situations where a group of people, wants to find things that the whole group likes. There are various applications where group recommenders can be applied specifically domains in which several people participate in a single activity. One example is the music that people listen in a bar/ club or at a restaurant. It must be chosen in a way that would satisfy most of the customers. One of the earlier group recommender systems was Polylens (M. O'Connor, D. Cosley, et al. 2001), based on the collaborative filtering approach, while more recent approaches use also other techniques, like the critique based approach of McCarthy et al (K. McCarth, M. Salamó, et al. 2006).

Jameson (A. Jameson 2004) identified four components that raise issues concern a group recommender System, as shown in **Table 2.7-1**:

Components	Issues raised
Members specify their preferences	What benefits and drawbacks can such examination have, and how can it be supported by the system?
The system generates recommendations	How can the aggregation procedure effectively discourage manipulative preference specification?
The system presents recommendations to the members.	How can relevant information about suitability for individual members be presented effectively?
Members decide which recommendation (if any) to accept.	How can the system support the process of arriving at a final decision when members cannot engage in face–to–face discussion?

Table 2.7-1: *Components and issues rose concerning a group recommender System*

Decision making within a group can be a long and complicated process. Group decisions are more complex than individual decisions. Decision makers may have different arguments in favor or against alternatives and any conflict among decision makers need to be overcome to achieve consensus and compromise when two or more people are involved in a decision making process. New recommendation techniques for group recommender systems need to be examined and designed accordingly to consider all the aspects of a group decision support system and how it can be merged into a Recommender System.

2.7.2 User oriented Recommender Systems

User oriented Recommender Systems focus on human special need and give more attention to human personality, sometimes by losing in system automation. However, Marcus Stolze, an IBM Watson Research Center researcher has talked about recommendation theory in the context of a "needs oriented system". Needs oriented systems are characterized by providing recommendations based on "human" needs. For example, the person may want a camera of a certain size or capable of fitting a certain lifestyle (perhaps it's good for sports photography). The user may or may not know which feature set matches their needs, and may not be able to articulate their needs in terms of features. New Recommender Systems need to support a focused understanding of customer needs. To be able to offer high quality assistance in complex product domains, Recommender Systems have to move away from their usually strictly feature-centric recommendation approaches, which focus on discrete features of an item, towards customer-oriented models.

2.7.3 Knowledge based recommendations

Alexander Felfernig on his talk on Knowledge based recommendation technologies at the "Recommenders06 Summer School on the Present and Future of Recommender Systems" pointed out that this type of systems was another highlight of the "real world" section of the recommendation talks. Alexander Felfernig focused on a number of application areas (such as financial services) that include "deep domain knowledge". These areas regularly rely on trained individuals with several years (or decades) of experience. Passing on the requisite knowledge of "what to do in situation x" is rarely a seamless process. A knowledge based recommendation system improves the training, maintenance, and "error handling" capacities of general customer/client management systems. These systems can be employed in sales, financial services, e-government, tourism, and computer support centers, virtually any field where expert advice and decision making is employed on a medium to large scale. Felfernig also provided supporting evidence of improved customer and user experience in the knowledge based recommendation systems. These results echoed earlier results in this field, underlining the potential for such systems across a broad range of fields.

2.7.4 Mobile Recommender Systems

Mobile Recommender Systems are systems that help a mobile user or a group of mobile users with decisions that they encounter 'on the go.' Examples include, consumers making purchasing decisions in retail stores, travelers searching for attractions, restaurants e.t.c. Recommender systems on wireless mobile devices may have significant impact on the way people shop in stores for example. However, very few studies have been made on designing recommender systems for mobile users. One such system is called MobyRek (F. Ricci and Q. N. Nguyen 2007) and approaches the recommendation problem by exploiting critique based techniques. The system has been designed to run on a mobile phone.

2.7.5 Tag Recommender Systems

The recent advent of the Web 2.0, which among others facilitates interactive information sharing, has been accompanied by an explosion in the number and of social content sites such as social tagging sites. The richness of information within social tagging sites presents a unique opportunity for the design of semantically-enriched recommender systems. Algorithms combining tags with recommenders may deliver both the automation inherent in recommenders, and the flexibility and conceptual comprehensibility inherent in tagging systems. Amer-Yahia et al. (S. Amer-Yahia, A. Galland*, et al.* 2008), propose a platform to incorporate multiple recommendation strategies depending on the user behavior and demonstrate it via x.qui.site, a system which gracefully incorporates user behavior into recommendations. For example, when a new URL is recommended to a user, it can be because many friends of the user are tagging it or because it is related to some tags that the user previously used. To this end, new similarity measures can be proposed to create user neighborhood of similar tagging behavior analogously to neighbors defined by similarity measures that exploit implicit ratings. Sen et al., also present tagommenders (S. Sen, J. Vig, et al. 2009), recommender algorithms that predict users' preferences for items based on their inferred preferences for tags.

Another dimension, under which tag recommenders can be developed, is not by exploiting user tags to identify similar users or provide recommendations, but to suggest a set of relevant keywords for the resources to be annotated.

2.7.6 Temporal and sequential Recommenders

Since user preferences vary over time modeling temporal dynamics should be a key when designing recommender systems or general customer preference models. Despite the high impact of temporal effects on user preferences, however, the subject attracted a quite negligible attention in the recommender literature. So far, typical recommender systems adopt a static view of the recommendation process in which the sequence of time plays no role at all, in predicting values.

Zimdars et al (A. Zimdars, D. M. Chickering, et al. 2001) considered collaborative filtering as a univariate time series prediction problem, and represent the time order of a user's votes explicitly when learning a recommendation model. They presented two techniques for transforming data that allow the collaborative filtering problem to be treated as a time-series prediction task.

Later on, Brafman et al. (R. I. Brafman, D. Heckerman, et al. 2003), extended this idea by stating that recommendation can be seen not only as a sequential prediction problem but also as a sequential decision problem and built their recommendation model based on Markov Decision Processes (MDP's).

Sugiyama et al. (S. Kazunari, H. Kenji, et al. 2004) proposed a personalized web search engine, where they let the user profile evolve over time. There, they distinguish between aspects of user behavior computed over a fixed time decay window, and ephemeral aspects captured within the current day.

Koren (Y. Koren 2009), by tracking the temporal dynamics of customer preferences in two different methods, a factorization model method and an item-item neighborhood model method, proved that the inclusion of temporal dynamics is very useful in improving quality of predictions, setting thus the foundations for the further development of temporal Recommenders.

Sequential Recommenders concern sequences of actions. For example sometimes it is natural to recommend actions that together will form a sequence (e.g., visits to stores in a shopping center), where the choice of the later actions will depend on the results of the actions earlier in the sequence (e.g., whether a desired product was found in the first store visited). This paradigm requires different recommendation techniques and raises new interface design issues.

2.7.7 Ubiquitous Recommender Systems

Ubiquitous computing systems with knowledge of more than locations, for example, the tools a person is using, could greatly benefit that person by recommending others who have expertise with those tools. For example, imagine if a device such as an LCD projector could learn best known methods and supply this information to users. When other people hooked their IBM ThinkPad to this projector, they selected low resolution.

"…Although it may be impossible to perfectly anticipate each individual's needs at any place or time, ubiquitous computing will enable such systems to help people cope with an expanding array of choices", David W. McDonald stated (D. W. McDonald 2003)

2.7.8 Nature inspired Recommender Systems

The use of the social insect colony behavior to solve computer problems such as combinatorial optimization, communications networks or robotics, has gained significant interest in recent years (J. Kennedy, R. C. Eberhart, *et al.* 2001). Social insects organized in colonies, e.g. ants, termites, bees, and wasps distinguish themselves by their organization skills without any centralized control. For example, Lorenzi *et al.* (F. Lorenzi, D. S. d. Santos, et al. 2005), presented the CASIS Recommender System in which they combined a case based reasoning approach with a metaphor from colonies of social insects, namely the honey bee dance which is used in nature to indicate the best nectar source among honey bees. Similarly they used it to retrieve the most similar case to a user's query.

Other types of nature inspired Recommenders are those that mimic a humanoid natural activity such as the way our immune system works. Xun et al.(Y. Xun and L. Quan-zhong 2007), presented a work inspired by the adaptive and self-organizing immune nature applied to the task of rating-based recommendation technology. Unlike the traditional vector-space model, user's preference model is represented in this case by the model of antigen and antibody. The artificial immune networks model, which is a type of competitive learning algorithm, is capable of extracting relevant features contained in antigens, and thus predictions and recommendations are made from the memory antibody cells which represent an "internal image" of the antigens. Acilar et al. (A. M. Acilar and A. Arslan 2008), tackle the two issues of Recommender Systems, the data sparsity and scalability by proposing a method of collaborative filtering algorithm which utilizes the Artificial Immune Network Algorithm.

2.8 Conclusions

Although Recommender Systems enumerate no more than 20 years of existence, they've encountered an exponential growth of research attention. These systems were rapidly developed both in research and commercial sector primarily due to their increased commercial interest.

From the first pilot Recommender System to the more recent and sophisticatedly developed, many approaches have been proposed, two of the most common being that of content-based and collaborative filtering. However, still today, no generic approach exists on Recommender Systems, making them thus an independent, active research area.

In this chapter a brief history on Recommender Systems together with the reference on the most popular Recommender Systems, may serve as a guide to a newcomer in this field, to obtain a complete image of Recommender Systems' past and current state.

Seven basic methodological current approaches in Recommender Systems were concisely mentioned in 2.4 to ensure an integrated and straightforward presentation of the major types of Recommender Systems that are nowadays met throughout literature.

This chapter closes with a brief mention on new trends in Recommender Systems to emphasize the importance and broadness of research in these types of systems.

3 User profiling and modeling

Contents

3.1 Introduction

The idea that software tools could act as filters of information and make selections on behalf of the user was generated as a natural consequence as soon as computer users felt frustrated by the plethora of information that was spread through the internet and the World Wide Web. Therefore it seemed natural to apply information processing power to the task of selecting items of interest and relevance. Nicholas Negroponte, founder and director of the Media Laboratory at Massachusetts Institute of Technology, one of the pioneer laboratories of Human-Machine research, is frequently referred as technology visionist. In his book *Being Digital* (N. Negroponte 1996), Negroponte presents his vision for the future of mediating technologies, how they could and should evolve and how they will impact our lives. Inter alia he refers to "digital butlers", software applications that will be endowed with enough intelligence to be knowledgeable about the user's preferences, interests, demands, e.t.c.

For software applications to act as information agents certain knowledge about the user is essential, the magnitude and types of which, highly depend on the role that the specific application serves. This knowledge about user preferences, behavior, acting style and more, is stored in the so called user profiles via a user modeling process. To design and develop a successful Recommender System, careful consideration of how to construct the so called user profile or model must be taken. In other words, user modeling is considered as an intermediate step of Recommender Systems, which in turn, share the same goal with personal information agents. The difference of personal information agents and Recommender System mostly lies in their different application areas.

Both user profiling and modeling are discussed in this chapter for two main reasons. First, user modeling is considered a crucial step of the proposed methodology and second, there is an inherent need for all related text books to clarify the difference between user modeling and user profiling since both terms are often used interchangeably.

3.2 Early stages and definitions

From the early times of user modeling that are attributed to the works of Allen (R. B. Allen 1990) and Elaine Rich (E. Rich 1983), severe advances have been made mostly in the direction of individualizing the user modeling process from system's other components. Since early stages, numerous application systems were developed that individualized user modeling and several review and survey papers appeared in the literature, establishing in this way user modeling research as an independent research field (G. Fischer 2001), (S. Stewart and J. Davies 1997), (A. Kobsa 2001). Initial user modeling systems collected different types of information and exhibited different kinds of adaptation to their current users.

User modeling is not a “young” topic but has gained popularity in the last years on one side because of its plethora of applications and on the other side due to the fast spreading out of Recommender Systems.

User modeling can be considered as a subfield of Human-Computer Interaction (HCI), a research field that in the first years of its history, focused on interfaces, particularly on graphical user interfaces (GUIs), for example using windows, icons, menus, and pointing devices to create more usable systems. As interface problems were evolved and further studied, the primary HCI concerns started to shift beyond the interface to include new essential challenges like improving the way people use computers to work, think, communicate, learn, critique, explain, argue, debate, observe, decide, calculate, simulate, and design (G. Fischer 2001).

Upendra Shardanand and Pattie Maes, who was named one of the "100 Americans to watch for" in the year 2000 by the Newsweek magazine, in their work “Social Information Filtering: Algorithms for Automating word of Mouth'' presented Ringo (U. Shardanand and P. Maes 1995), which makes personalized recommendations for music albums and artists. Ringo was one of the first applications that introduced social filtering, an ancestor of the collaborative filtering technique that is widely applied in Recommender Systems (J. B. Schafer, D. Frankowski, *et al.* 2007). Ringo was also one of the first demonstrations of the common sense fact that people's tastes are not randomly distributed. Instead there are general trends and patterns within the taste of a person and as well as between groups of people. It raised therefore the point of using profiles to represent users. In Ringo, people described their listening pleasures to the system by rating some music. These ratings constituted the person's profile. The system then compared this profile to the profiles of other users, and weighted each profile for its degree of similarity with the user's profile. This profile also changed over time as the user rates more artists.

User modeling is a cross-disciplinary research field that attempts to construct models of human behavior within a specific computer environment. Contrary to traditional artificial intelligence research, the goal is not to imitate human behavior as such, but to make the machine able to understand what the expectations, goals, information needs, desires etc. of a user are, in terms of a specific computing environment. Furthermore, the goal is to utilize this understanding to assist the user in performing computing tasks. The computer representation of the user's goals is called a user model and systems that construct and utilize such models are called user modeling systems.

The existence of discrete patterns among users does not preclude the fact that different users also have different knowledge, interests, abilities, learning styles, and preferences. This is the reason why traditional or typical user models have been proved insufficient. The necessity of considering user's individual needs and capabilities in their computer interaction task has been identified in early times by Elaine Rich (E. Rich 1983) who describes an example of conflicting requirements that point to the need for a system that can appear differently to different users. The example concerned one study of the performance of expert users at a text editing task suggesting that the number of keystrokes required to perform an operation should be minimized, in contrast to another study of people just learning to use an editor and the result of which suggests that English-like, full word commands should be used.

Since mid 90's when various user modeling techniques appeared sequentially the difficulty to determine the boundaries of a user model became apparent. In fact, this problem arose from the plethora of definitions attributed to the process of user modeling. Fischer (G. Fischer 2001) for example defines user models as: "*models that systems have of users that reside inside a computational environment*". It might be argued that all programs have an implicit user model, but not necessarily an adaptive model. Indeed programs may be developed that serve one user's needs in only one situation, rather than an integrated more general program. Further, user models, especially those which involve complex inferences, may be seen as a collection of models rather than a single model. Given these problems, a general but functional definition was proposed by Allen (R. B. Allen 1990): "*A user model is the knowledge and inference mechanism which differentiates the interaction across individuals*".

Modeling the user may be as simple as fitting a user profile (e.g., single, young, female) or as complicated as discovering expert knowledge (e.g., how a chemist would classify a data set). The terms user profiling and user modeling are often used interchangeably. The difference between user profiling, and user modeling, lies in their different level of sophistication. Depending on the content and the amount of information about the user, which is stored in the user profile, a user can be modeled. Thus, the user profile is used to retrieve the needed information to build up a model of the user. User profiling is simply seen as the process of collecting raw data about the user. The term user modeling has been applied not only to the process of gathering information about the users of a computer system but also of using the information to model the user in order to provide services or information adapted to the specific requirements of individual users (or groups of users). User models may represent cognitive styles, intellectual abilities, intentions, learning styles, preferences and interactions with the system.

3.3 Traditional User modeling methodologies and techniques

There are different aspects of User models. Predictive models for example that provide helpful indications of how people will approach and perform tasks constitute a single simplified type of user models. On the other hand, both the efficiency with which information can be accessed from the worldwide web and the satisfaction of the user in doing so can be enhanced by adapting web tools to individual users, taking into account their different preferences, knowledge, and goals. User models that handle preference information are extremely effective in helping users find what they are looking for and constitute a more sophisticated type of user models.

There are many different techniques that are used to build user models and this makes their classification an intricate process. User models have been classified in various aspects by different researchers that either overlap or not. For example in (H. Lyer 1998) user models are classified as analytical or quantitative, process or non process and dynamic or static. On the other hand Elaine Rich (E. Rich 1983) classifies user models as:

- One model of a single, canonical user vs. a collection of models of individual users.

- Models specified explicitly either by the system designer or by the users themselves vs. models inferred by the system on the basis of users' behavior.

- Models of fairly long-term user characteristics such as areas of interest or expertise vs. models of relatively short-term user characteristics such as the problem the user is currently trying to solve.

A taxonomy framework is proposed in this thesis instigated by the different classification frameworks proposed in the literature that identifies five main dimensions under which a user model can be characterized. These are: **a)** *user identification*, **b)** *profile information acquisition*, **c)** *user modeling temporality*, **d)** *user modeling memory* and **e)** *user modeling representation and learning*. These dimensions are in no case mutually exclusive, they can rather be considered as an attempt for the formation of a redundant set of dimensions under which existing user modeling systems can be classified. The dimensions identified here also exhibit a hierarchical nature as discussed shortly. The five dimensions of the proposed classification scheme distinguish user models according to the following components:

3.3.1 User identification

There are five basic approaches to user identification: software agents, logins, enhanced proxy servers, cookies, and session ids (S. Gauch, M. Speretta, et al. 2007). Software agents, logins and enhanced proxy servers are more accurate, but they also require the active participation of the user, for example software agents require user-participation in order to install the desktop software. It is considered though the most reliable method because there is more control over the implementation of the application and the protocol used for identification. The next most reliable method is based on logins. Because the users identify themselves during login, the identification is generally accurate, and the user can use the same profile from a variety of physical locations. On the other hand, the user must create an account via a registration process, and login and logout each time they visit the site. Enhanced proxy servers can also provide reasonably accurate user identification. However, they require that the user register his/ her computer with a proxy server. Thus, they are generally able to identify users connecting from only one location, unless users bother to register all of the computers they use with the same proxy server. The final two techniques, cookies and session ids, are less invasive methods. The first time that a browser client connects to the system, a new user-id is created. This id is stored in a cookie on the user's computer. When they revisit the same site from the same computer, the same user-id is used. This places no burden on the user at all; however, if the user uses more than one computer, each location will have a separate cookie, and thus a separate user profile. Also, if the computer is used by more than one user, and all users share the same local user id, they will all share the same, inaccurate profile. Finally, if the user clears their cookies, they will lose their profile altogether, and if users have cookies turned off on their computer, identification and tracking is not possible. Session ids are similar, but there is no storage of the user-id between visits. Each user begins each session with a blank slate, but their activity during the visit is tracked. In this case, no permanent user profile can be built, but adaptation is possible during the session.

3.3.2 Profile information acquisition

The modeling system may acquire information explicitly, relying on personal information input by the users, or by means of a user-completed questionnaire, or implicitly, by observing user actions and making inferences based on stored knowledge. Although explicit information is generally considered more accurate, explicit user information acquisition has the drawback of time cost and burden for the user and requires the user's willingness to participate. Many commercial sites, such as google.com, offer customization by asking the user to provide specific preference information. igoogle (http://www.google.com/ig) for instance, is a customizable page launched by Google that lets you create a personalized homepage that contains a Google search box at the top, and your choice of any number of gadgets below. This is an example of explicit information provided by the user. This service provides a simple type of personalization that is referred as customization and is solely based on the explicit information provided by the user. Customization though may be considered a simple form of personalization and is differentiated is the sense that personalization goes further in depth by providing the user with personalized recommendations rather than simply adapt to user manipulated information.

Implicit data acquisition has the advantage of not requiring any additional intervention by the user. Also, user model design can include both *implicit* and *explicit* acquisition methods (E. Shifroni and B. Shanon 1992). Papazoglou (M. P. Papazoglou 2001) for example uses an automatic component to build a user profile based on user observations, but he also provides a mechanism for explicit relevance feedback in order to better tailor the profiles to user's individual interests.

Examples of explicit data collection include the following:

- Asking a user to rate an item on a sliding scale.
- Asking a user to rank a collection of items from favorite to least favorite.
- Presenting two items to a user and asking him/her to choose the best one.
- Asking a user to create a list of items that he/she likes.

Examples of implicit data collection include the following:

- Observing the items that a user views in an online store.
- Analyzing item/user viewing times.
- Keeping a record of the access logs where entries correspond to HTTP requests typically containing the client IP address, time-stamp, access method, URL, protocol, status and file size
- Obtaining the history of the user requests for current and past browsing sessions that is maintained by most browsers i.e. a list of items that a user has listened to or watched on his/her computer.
- Analyzing the user's social network and discovering similar likes and dislikes.

- Analyzing bookmarks that provide a fast way of accessing a set of documents representing user interests.

Data acquisition is just the first step in creating a user model. Evidently, data acquisition is the primer step and even if it may be considered as the simplest one during the user modeling process, it constitutes an undoubtedly crucial step since it attains a substantial impact on possibilities of the subsequent user modeling stages. A common saying in the field of computer science goes like: "Garbage In, Garbage Out" also abbreviated as GIGO. It is used primarily to call attention to the fact that computers will unquestioningly process the most irrational of input data and produce irrational output. In the same sense if the user modeling algorithm receives insufficient or incorrect information, no matter the sophistication level of the algorithm, the output be will be surely useless for the application.

3.3.3 User modeling temporality

The goal of user modeling may be to predict user behavior, to gain knowledge of a particular user in order to tailor interactions to that user, or to create a database of users that can be accessed by others. User profiles are either static or dynamic. Their main difference is that a dynamic model would be altered during the course of the interaction or at the end of a session whereas a static model, once constructed, would remain unchanged. Static information related to user modeling is information about the user that hardly changes in time, for example physical characteristics like gender, weight, length etc. As people's behavior and preferences change over time, static profile degrades in quality over time. Dynamic model can be described as a model reflecting changes of state dependent on interaction with the system and static model as a model embodying permanent states of the user. Dynamic profiling can be done by:

- *Monitor User's Actions*. Browsing patterns and clicking activity in the interface provide another source of information about users. Such activity is analyzed to determine topics and concepts of interest through off-line data mining.

- *Monitor User Search History.* Most users will not remember the whole process of their search and how they arrived at the results they wanted but rely on keywords or nodes to help them recollect their search routines.

3.3.4 User modeling memory

The discrimination of short or long term user models is made according to the volatility of the information kept about. While long term facts are kept for subsequent interactions of the user with the system, short term facts may be safely deleted at the end of this interaction. In order to distinguish the difference between short and long term user models and to point out their relative importance Rich (E. Rich 1983) mentions the following example:

Consider a system of travel booking flights. A possible dialogue in such a system could be like:

> ***Customer:*** How much is a ticket to New York?
>
> ***Clerk:*** One hundred dollars.
>
> ***Customer:*** When is the next plane?
>
> ***Clerk:*** The next plane is completely booked, but there's still room on one that leaves at 8:04.
>
> ***Customer:*** O.K., I'll take it.

An appropriate user model for a system like the aforementioned would only exploit short term information about the user, the fact that he/she plans to travel to New York. But what if the same customer appears the next day in New York and uses an information system to get recommendations for museums or places to visit. In this case the model should contain information of long term i.e. preferences, hobbies e.t.c. to be able to accurately propose places of interest for the specific user.

3.3.5 User modeling learning and representation

User models are constructed to represent user preferences. Preferences provide a means for specifying desires in a declarative way, which is a point of critical importance for Artificial Intelligence (AI). Drawing on past research on knowledge representation and reasoning, AI offers qualitative and symbolic methods for treating preferences that can reasonably complement hitherto existing approaches from other fields, such as decision theory and economic utility theory. Needless to say, however, the acquisition of preferences is not always an easy task. Therefore, not only are modeling languages and representation formalisms needed, but also methods for the automatic learning, discovery, and adaptation of preferences.

Many applications require building user models that not only perform as humans do, but also learn in order to predict human behavior in an upcoming event. Most existing user models are developed for specific tasks, or aspects of tasks (e.g., menu search, icon identification, deployment of attention, or automatization), and then validated on a case-by-case basis using human data (A. Johnson and N. Taatgen 2004).

Stewart and Davies (S. Stewart and J. Davies 1997) review user modeling techniques by classifying them into three main categories: a) Statistical Term-based approaches that consider well established techniques from the information retrieval field, such as Term Matching or Latent Semantic Indexing, b) Artificial Intelligence (AI), and Neural Networks (NN) approaches in which user profiles are updated by agents that learn user preferences through feedback learning information and c) Social Filtering approaches that enable a user to filter the information that they receive based upon the ratings given by other users in the system.

Since then numerous other techniques have been used in the user modeling learning process such as:

Statistical and Probabilistic methods:

The statistical term based approaches are probably the most common, since they are relatively straightforward to implement. There has been a lot of research into methods for improving the performance of the simple term matching techniques and very efficient algorithms have been developed.

Term matching is one of the simplest ways of determining whether some information is relevant to a user's interests. A common method for implementing a term matching scheme is to use a Vector Space Model. Such a model constructs a vector of dimension *m* for each document, where *m* is the total number of terms used to identify the content of the documents in the system.

The *TF-IDF* technique is one of the most successful approaches for deriving the vector representation of a particular document, in which the weighting given to a term depends on the Term Frequency, (TF), and the Inverse Document Frequency, (IDF). For details on this method can be found in section 2.4.1 of Chapter 2.

Support Vector Machines use vector similarity to measure how similar a content is, to the contents selected by a given user (i.e., a user history). The problem is that those representations of preference do not always correspond to a common sense of preference. First, when preference is represented by vector similarity, it is hard to get intuitive interpretation about how much a user likes or dislikes a given item.

Affinity analysis is the study of attributes or characteristics that "go together." Methods for affinity analysis, seek to uncover associations among these attributes; that is, it seeks to uncover rules for quantifying the relationship between two or more attributes. Association rules take the form "If antecedent, then consequent," along with a measure of the support and confidence associated with the rule. For example, a particular supermarket may find that of the 1000 customers shopping on a Thursday night, 200 bought potato chips, and of the 200 who bought potato chips, 50 bought beer. Thus, the association rule would be: "If buy potato chips, then buy beer," with a support of 50/1000 = 5% and a confidence of 50/200 = 25% (D. T. Larose 2005).

In association rule mining, preference is represented by the strength of association (i.e., correlation) between an item and a user history. The current association measures can be inappropriate to represent preference because they were originally designed to discover useful rules rather than preferences.

Probabilities are used to predict user's future behaviors in a Bayesian network for example, where preference is represented by the probability that a user selects a given item. When the preference is represented by the probability that a user selects an item, a preferred item with low frequency, or a not so preferred item with high frequency, is not measured correctly. It happens because the probability of user selection is determined not only by his/her preference, but also by the accessibility of an item. For precise representation of preference, the actual concept of preference should be separated from that of probability.

Jung et al. (S. Y. Jung, J.-H. Hong, *et al.* 2005) proposed a statistical preference model using mutual information and combining information of joint features using random occurrence probability to alleviate problems raised by data sparseness. They advocated that the probability that a user selects an item (or feature) is determined mainly by two factors: the preference on the item and the accessibility of the item.

Xu et al. (G. Xu, Y. Zhang, *et al.* 2005) propose a Web recommendation framework based on user profiling technique. The usage pattern knowledge, in the form of user profile derived from Web usage mining, is combined into Web recommendation system to improve the efficiency of recommendation by predicting user-preferred content and customizing the presentation. During pattern discovery stage, probabilistic inference method based on Probabilistic Latent Semantic Analysis (PLSA) model, a variant of LSA, is exploited to model the underlying relationships among the co-occurrence activities and identify the latent task model in terms of latent semantic factor. The PLSA model is based on a statistic model called aspect model, which can be utilized to identify the hidden semantic relationships among general co-occurrence activities.

Wong and Butz (S. K. M. Wong and C. J. Butz 2000) suggest the use of Bayesian networks for user profiling and propose a method for constructing a user profile using either a Bayesian or Markov network. The input to their approach is a sample of documents that the user has marked as either relevant or non-relevant. The user profiles then learn a probabilistic network which encodes the user's preferences. Such a network provides a formal foundation for probabilistic inference. Documents can then be ranked according to the conditional probability defined by the network.

Clustering methods

Cluster Analysis is commonly performed to create user profiles. Cluster Analysis divides data into groups (clusters) that are meaningful, useful, or both (P.-N. Tan, M. Steinbach, *et al.* 2006). It is an unsupervised process aiming at grouping data objects, based only on information found in the data that describes the objects and their relationships. The goal of cluster analysis is that the objects within a group be similar (or related) to one another and different from (or unrelated to) the objects in other groups.

In a preliminary work for the development of the system that is analytically described in this thesis the role of representation of user profiles in order to create user profile groups was studied (K. Lakiotaki, P. Delias, *et al.* 2009). There, it was shown that the incorporation of a Multi-criteria methodology prior to the application of a clustering algorithm constitutes a fundamental step for the formation of more accurate user profiles in terms of compactness and separation of the groups. More analytically, it was proved that clustering customers according to their preferences leads to better results when utility functions are used to represent preference information, as compared to bare ranking orders stated by the customers. The framework under which the aforementioned statement was proved is shown in **Figure 3.3-1**.

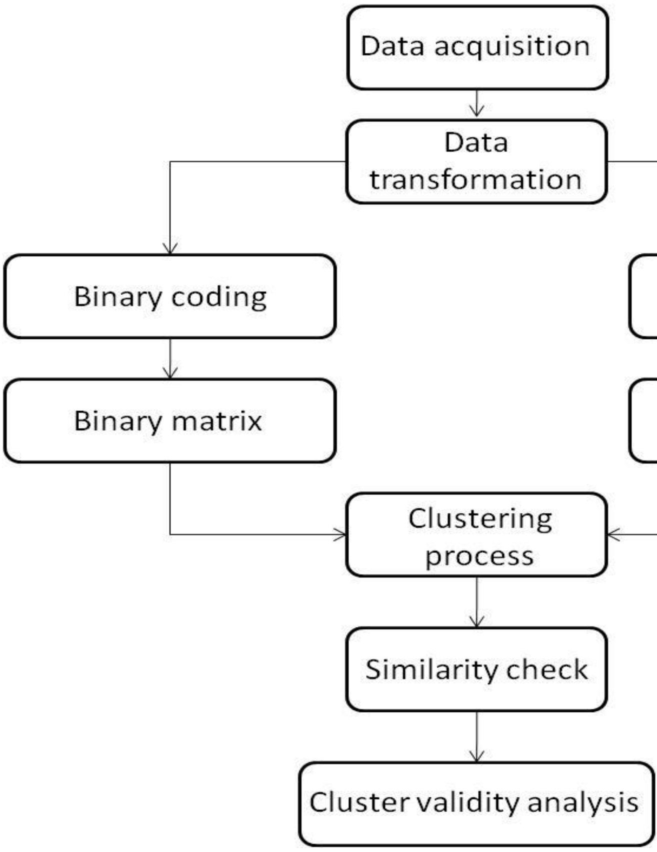


Figure 3.3-1: *Methodological framework under which the superiority of user profiling representation by utilities compared to ranking orders was proved.*

The focus of this work was to elucidate the role of utility functions in discovering groups of people with similar preferences. It started with the assumption that utility information representing customers' preferences is essential for the formation of more accurate groups of customers. This accuracy was measured in terms of more dense and furthest separated clusters. This hypothesis was mathematically proved in the in that paper by exploiting cluster validity analysis. The results presented proved that taking into account the information provided by the utility functions, as calculated by a Multi-criteria analysis algorithm during the clustering phase, led to more homogeneous groups of people than dealing with ranking orders (as stated in a questionnaire). Furthermore, this conclusion was verified by the concordance of the results of the two different clustering techniques used. Both the global k-means algorithm and the Agglomerative Hierarchical Clustering technique concluded that the clusters formed by considering the utility functions are better, in terms of compactness and separation of the groups and thus learning and representing clusters by rich preferential information is a crucial step in user modeling.

Godoy and Amandi (D. Godoy and A. Amandi 2006) presented a document clustering algorithm named WebDCC (Web Document Conceptual Clustering), that carries out incremental, unsupervised concept learning over Web documents in order to acquire user profiles.

Fuzzy clustering has been also exploited in user profiling techniques. In such kind of clustering a user can be at the same time in more than one cluster with different degrees of membership. This allows to better capture the inherent uncertainty that the problem of modeling user behavior has. Examples of applications that implement a recommendation task using FC include (T. Lampinen, M. Laurikkala, et al. 2005) and (O. Nasraoui, H. Frigui, et al. 1999).

Classification methods

Decision trees can be used to classify users and/or documents in order to use this information for personalization purposes. Decision trees can also handle noisy data and/or data with missing parameters, which makes them very useful for creating user models due to the noisy and imprecise nature of the data available (E. Frias-Martinez, S. Y. Chen, et al. 2006).

A decision tree is a collection of decision nodes, connected by branches, extending downward from the root node until terminating in leaf nodes. Beginning at the root node, which by convention is placed at the top of the decision tree diagram, attributes are tested at the decision nodes, with each possible outcome resulting in a branch. Each branch then leads either to another decision node or to a terminating leaf node.

Other classification algorithms such as C4.5 have also been used in user modeling. Paliouras et al. (G. Paliouras, V. Karkaletisis, *et al.* 1999), for example build stereotypes by exploiting supervised learning (C4.5) on personal data extracted from a set of questionnaires answered by the users of a news filtering system.

The C4.5 algorithm is Quinlan's extension of the ID3 algorithm for generating decision trees (J. R. Quinlan 1992). The C4.5 algorithm recursively visits each decision node, selecting the optimal split, until no further splits are possible. The algorithm uses the concept of information gain or entropy reduction to select the optimal split.

A survey of data mining techniques within the area of user modeling can be found in and there, more classification methods used to create user models are discussed (E. Frias-Martinez, S. Y. Chen, et al. 2006)

Relevance feedback methods

The basic idea behind relevance feedback in information filtering systems for example is to exploit user information that is passed to the systems as *feedback*, to discover *relevance* information to show to this user. Based on the user's query and the document corpus, possible contexts for the query are inferred and used to suggest additional query terms. In INVAID for instance (L. Kelly and J. Dunnion 1999), the system receives explicit user feedback through ratings of relevant pages and suggests pages of interest to users based on the feedback of the user coupled with filtering strategies. Pazzani et al and Aniscar and Tasso created intelligent agents that will analyze user feedback based on ratings defined by the user on the visited page as a measure of user interest. They perform an extended navigation of related pages and graphically show the set of the pages found, classified according to the user's interest. These systems request users to provide explicit feedback on documents in terms of ratings or preferences. Stewart et al (S. Stewart and J. Davies 1997) created intelligent agents that analyze the user feedback based on well defined ratings of visited pages as a measure of user interest. All the above systems request users to provide explicit feedback on items. These methods are commonly referred to as relevance feedback. The general principle is to allow users to rate items returned by the retrieval system with respect to their information need. This form of feedback can subsequently be used to incrementally refine the initial query. In a manner analogous to rating items, there are explicit and implicit means of collecting relevance feedback data.

Rocchio's algorithm is the classic algorithm for implementing relevance feedback (G. Salton 1971). It models a way of incorporating relevance feedback information into a vector space model. The algorithm is based on the adjustment of an initial query through differently weighted terms of relevant and non-relevant documents. The approach forms two document categories by taking the vector sum over all relevant and non-relevant documents. The following formula summarizes the algorithm formally:

$$Q_{i+1} = aQ_i + b\sum_{rel}\frac{D_i}{|D_i|} - g\sum_{non-rel}\frac{D_i}{|D_i|} \qquad \textbf{3.3.5-1}$$

In 3.3.5-1 Q_i is the user query at iteration i and a, b, g are parameters that weight the initial query and the two TF-IDF terms accordingly.

3.4 User modeling based on Multiple Criteria Decision Analysis

3.4.1 Brief introduction

Almost every aspect of our everyday life, from the simplest to the most complicated, involves decisions. The history of decision making is long, rich, and diverse. From the prehistoric times when human decisions were guided by interpretations of entrails, smoke, dreams etc, to recent times when human decision making is modeled and computerized systems have been developed to support decision making, human life has been marked and driven by decisions. Nowadays, neuroscientists are mapping the risk and reward systems in the brain that drive our best and worst decision making and new scientific disciplines of decision theory such as Neuroeconomics (P. W. Glimcher, C. Camerer, *et al.* 2008) that has its origins in events following the neoclassical economic revolution of the 1930s, and the birth of cognitive neuroscience during the 1990s, prove that although decision theory is as old as humanity, it is still considered an open research field. A brief yet complete history of Decision Making can be found in review of (L. Buchanan and A. O'Connel 2006). Modern Decision Theory has been developed since the middle of the 20th century through contributions from several academic disciplines and researchers who identify themselves as economists, statisticians, psychologists, political and social scientists or philosophers.

In Decision Theory, the field of Multiple Criteria Decision Analysis (MCDA) emerged from the fact that real world decision making problems are intrinsically multidimensional. Practical problems are often characterized by the existence of multiple, often conflicting criteria. Multiple Criteria Decision Analysis aims at giving the decision maker a recommendation, in other words aiding the decision maker in the so called decision making process, concerning a set of objects, actions, alternatives, items etc, evaluated on multiple points of view, which are roughly referred as criteria (attributes, features, variables etc). The field of Multiple Criteria Decision Analysis (MCDA) is a well established field of Decision Science, and comes into a large variety of theories, methodologies, and techniques(J. Figueira, S. Greco*, et al.* 2005).

Multiple Criteria Decision Analysis aims at assisting a decision maker in dealing with the ubiquitous difficulties in seeking compromise or consensus between conflicting interests and goals, represented by the roughly referred as "multiple criteria". A common approach states that Multiple Criteria Decision Analysis is a methodology enabling the construction of a reliable and convincing model when several alternatives need to be assessed against multiple attributes under different problem statements (choosing, ranking, classifying etc.). In abstract, we may principally claim, that the word "aiding" in MCDA implies that the decision model representing decision maker's value system "supports" and in no case substitutes the decision maker during the decision process.

In this section some principal aspects of Decision theory under the Multiple Criteria Analysis perspective are discussed as well as the underlying attributes that affect the design and development of a Recommender System that incorporates MCDA principles at some point.

Already in the 18th century, Condorcet (1743-1794), divided the decision process into three stages, the "first discussion phase", where the principles that would serve as the basis for the decision are discussed, the "second discussion phase", in which the question is clarified, opinions approach and combine with each other to a small number of more general opinions and also the alternatives are determined and the “third phase”, which consists of the actual choice between these alternatives.

Later on, Simon (H. A. Simon 1977) adjusted the existent approaches to become suitable for decisions in organizations into three phases, the “intelligence”, the “design” and the “choice” phase.

3.4.2 The disaggregation-aggregation approach

As already stated, in decision-making involving multiple criteria, the basic problem stated by analysts and decision-makers concerns the way that the final decision should be made. Two basic approaches under which numerous methodologies have been developed are a) *the bottom up approach* and b) *the top-down approach*. The first, the most trivial one, includes all methodologies and techniques that aim at building a decision model by aggregating preferential information on criteria. This means that the criteria aggregation model is known a priori and the analyst guides the decision maker via a process of incrementally declaring preferences to construct his/ her global preference model. This approach is referred to the literature as the aggregation approach and several representatives of it include MAUT, SMART, TOPSIS, MACBETH, or AHP(J. Figueira, S. Greco*, et al.* 2005). The second approach, can be though as a reverse process in which the final decision in know a priori and is decomposed to reveal the underlying attributes that led the decision maker to the specific decision and utilize this information to construct a value system that will be used in future decision support cases.

The UTA (*UTilités Additives*) methods are considered the most representative methods of the second approach, also known in the literature as the Disaggregation-Aggregation approach (E. Jacquet-Lagreze and Y. Siskos 1982) or simply the Disaggregation approach. UTA methods refer to the philosophy of assessing a set of value or utility functions, assuming the axiomatic basis of MAUT and adopting the preference disaggregation principle. UTA methodology uses linear programming techniques in order to optimally infer additive value/utility functions, so that these functions are as consistent as possible with the global decision-maker’s preferences (inference principle). More details as well as a historical background on the development of the preference disaggregation philosophy can be found in the review of Jacquet-Lagreze and Siskos (E. Jacquet-Lagreze and Y. Siskos 2001).

Multiple Criteria Analysis can be defined "*a set of methods or models enabling the aggregation of multiple evaluation criteria to choose one or more actions from a set A*", or more generally as "*an activity of decision-aid to a well-defined decision-maker (individual, organization, etc.)*". Based on those concepts, various approaches have been developed.

In both cases however, the set *A* of potential actions (or objects, alternatives, decisions) is analyzed in terms of multiple criteria in order to model all the possible impacts, consequences or attributes related to the set *A*.

In early stages of MCDA, Roy (B. Roy 1985) outlined a general modeling methodology of decision-making problems, which included four modeling steps starting with the definition of the set *A* and finishing with the activity of decision-aid. These are:

- Level 1: Object of the decision, including the definition of the set of potential actions A and the determination of a problematic on A.
- Level 2: Modeling of a consistent family of criteria assuming that these criteria are non-decreasing value functions, exhaustive and non-redundant.
- Level 3: Development of a global preference model, to aggregate the marginal preferences on the criteria.
- Level 4: Decision-aid or decision support, based on the results of level 3 and the problematic of level 1.

A more recent paper of Tsoukiàs (A. Tsoukiàs 2007), introduces a descriptive model of the "decision aiding process", that is the set of activities occurring between a Decision Maker and an Analyst who develops a formal model aiming at helping the Decision Maker to face a problem situation. This model considers the decision aiding process as a cognition process, introducing schematically the cognitive artifacts aiming at supporting the Decision Maker's decision process. Within such a model a *recommendation* results from the construction of an *evaluation model* resulting from a *problem formulation* which represents formally a specific *problem situation.*

At the second level of a decision process which involves the *problem formulation* three basic attitudes must be formulated:

- The first concerns constructing a set of feasible and realistic alternative actions A.
- The second concerns describing a set of actions under a set of precise instances of the points of view established in V, where V is the set of points of view under which the potential actions are expected to be observed, analyzed, evaluated and compared.
- The third consists of partitioning the set A.

Partitioning the set A implies to establish a set of categories to which each element of A will be associated. According to Tsoukias (A. Tsoukiàs 2007) eight possible problem statements, the six of which are considered operational, and the two "non operational" which are called "design" and "description", cover all possible problem statements that may arise during a decision process.

- Predefined, not ordered categories (a typical example being the assignment problem);
- Predefined, ordered categories (as in the "sorting" procedures);
- Two, not predefined, not ordered categories (as when we partition the elements of A in similar or analog objects and not);
- More than two, not predefined, not ordered categories (as in the clustering and more generally classification case);
- Two, not predefined, ordered categories (for instance the chosen or rejected objects and the rest);
- More than two, not predefined, ordered categories (as in the ranking procedures).

At the second level of *problem formulation*, the points of view *V* under which the potential actions will be analyzed and evaluated may conclude on a consistent family of criteria {g_1,g_2,...,g_n}. Each criterion is a non-decreasing real valued function defined on *A*, as follows:

$$g_j : A \rightarrow [g_{j*}, g_j^*] \subset \mathbb{R} \,/\, a \rightarrow \mathbf{g}(a) \in \mathbb{R} \qquad \textbf{3.4.2-1}$$

where $[g_{j*}, g_j^*]$ is the criterion evaluation scale, g_{j*} and g_j^* are the worst and the best level of the *i-th* criterion respectively, g(*a*) is the evaluation or performance of action *a* on the *i-th* criterion and **g**(*a*) is the vector of performances of action *a* on the *n* criteria.

From the above definitions the following preferential situations can be determined:

$$\begin{cases} g_i \;\; \alpha \;\; > g_i \;\; b \;\; \Leftrightarrow \alpha \succ b \; (\alpha \text{ is preferred to } b) \\ g_i \;\; \alpha \;\; = g_i \;\; b \;\; \Leftrightarrow \alpha \sim b \; (\alpha \text{ is indifferent to } b) \end{cases} \qquad \textbf{3.4.2-2}$$

So, having a weak-order preference structure on a set of actions, the problem is to adjust additive value or utility functions based on multiple criteria, in such a way that the resulting structure would be as consistent as possible with the initial structure. This principle underlies the disaggregation-aggregation approach presented in the next section.

In the traditional aggregation paradigm, the criteria aggregation model is known a priori, while the global preference is unknown. On the contrary, the philosophy of disaggregation involves the inference of preference models from given global preferences. The Disaggregation-Aggregation approach aims at analyzing the behavior and the cognitive style of the Decision Maker (DM) (E. Jacquet-Lagreze and Y. Siskos 2001). Special iterative interactive procedures are used, where the components of the problem and the DM's global judgment policy are analyzed and then they are aggregated into a value system as shown in **Figure 3.4-1**. The goal of this approach is to aid the DM to improve his/her knowledge about the decision situation and his/her way of preferring that entails a consistent decision to be achieved.

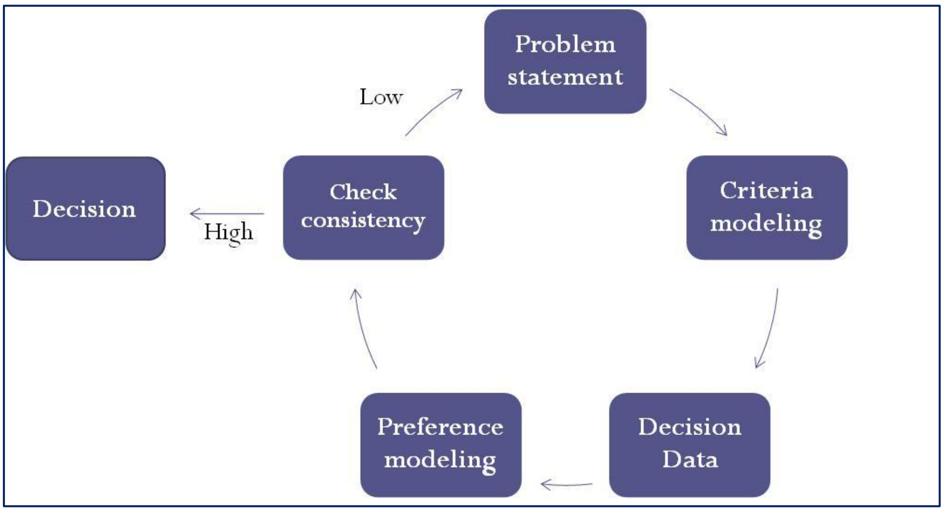


Figure 3.4-1: *The Disaggregation – Aggregation approach*

In order to use global preference given data, Jacquet-Lagrèze and Siskos (E. Jacquet-Lagreze and Y. Siskos 2001) note that the clarification of the DM's global preference necessitates the use of a set of reference actions A_R . Usually, this set could be:

- A set of past decision alternatives (A_R: past actions),
- A subset of decision actions, especially when A is large ($A_R \subset A$),
- A set of fictitious actions, consisting of performances on the criteria, which can be easily judged by the decision-maker to perform global comparisons (A_R: fictitious actions).

In each of the above cases, the DM is asked to express and/or confirm his/her global preferences on the set A_R taking into account the performances of the reference actions on all criteria.

The UTA *(UTilités Additives*) method proposed by Jacquet-Lagrèze and Siskos (E. Jacquet-Lagreze and Y. Siskos 1982) aims at inferring one or more additive value functions from a given ranking on a reference set A_R. The method uses special linear programming techniques to assess these functions so that the ranking(s) obtained through these functions on AR is (are) as consistent as possible with the given one.

The basic formulation under which UTA method was developed is described hereupon, to ensure a complete presentation of the mathematical formulation on which UTA*, a UTA based algorithm incorporated in the proposed methodology was developed.

The criteria aggregation model in UTA is assumed to be an additive value function of the following form:

$$u(\mathrm{g}) = \sum_{i=1}^{n} p_i u_i(g_i) \qquad \textbf{3.4.2-3}$$

subject to normalization constraints:

$$\begin{cases} \sum_{i=1}^{n} p_i = 1 \\ u_i(g_{i*}) = 0, \quad u_i(g_i^*) = 1, \quad \forall i = 1, 2, ..., n \end{cases} \qquad \textbf{3.4.2-4}$$

where u_i, $i=1,2,...,n$ are non decreasing real valued functions, named marginal value or utility functions, which are normalized between 0 and 1, and correspond to the weight of **Figure 3.4-2.**

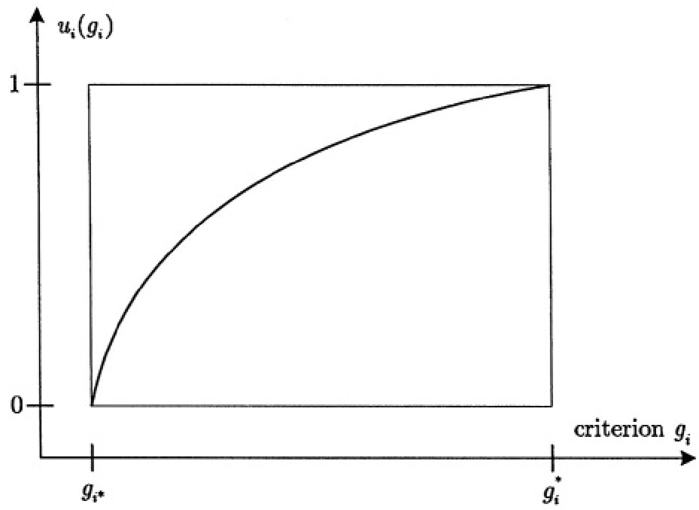


Figure 3.4-2: *The normalized marginal value function*

Both the marginal and the global value functions have the monotonicity property of the actual criterion. Very recently however Kliegr (T. Kliegr 2009) presented a study on the problem of non-monotonicity in UTA-methods. He developed a method called UTA-NM, which keeps, with the exception of the monotonicity, all basic principles of the UTA method. He introduces the concept of non-monotonicity in UTA through inclusion of an element, penalizing the model complexity into the objective function.

Independent on the monotonicity of preferences, in the case of the global value function the following properties must hold:

$$\begin{cases} u[g(a)] > u[g(b)] \Leftrightarrow \alpha \succ b \ (preference) \\ u[g(a)] = u[g(b)] \Leftrightarrow \alpha \sim b \ (indifference) \end{cases} \quad \textbf{3.4.2-5}$$

The UTA method infers an unweighted form of the additive value function, equivalent to the form defined from relations 3.4.2-3 and 3.4.2-4, as follows:

$$u(\mathrm{g}) = \sum_{i=1}^{n} u_i(g_i) \quad \textbf{3.4.2-6}$$

subject to normalization constraints:

$$\begin{cases} \sum_{i=1}^{n} u_i(g_i^*) = 1 \\ u_i(g_{i^*}) = 0, \quad \forall i = 1, 2, \ldots, n \end{cases} \quad \textbf{3.4.2-7}$$

The existence of such a preference model assumes the preferential independence of the criteria for the DM as defined by Kenney and Raiffa (R. Keeney and H. Raiffa 1993) On the basis of the additive model 3.4.2-6 and 3.4.2-7, and by taking into account the preference conditions 3.4.2-5, the value of each alternative may be written as:

$$u'[g(a)] = \sum_{i=1}^{n} u_i[g_i(a)] + \sigma(\alpha) \quad \forall \alpha \in \mathrm{A}_R \quad \textbf{3.4.2-8}$$

where *σ(a)* is a potential error relative to *u'[**g**(a)]*. Moreover, in order to estimate the corresponding marginal value functions in a piecewise linear form, Jacquet-Lagreze and Siskos (E. Jacquet-Lagreze and Y. Siskos 1982) proposed the use of linear interpolation. For each criterion, the interval is cut into equal intervals, and thus the end points are given by the formula:

$$g_i^j = g_{i*} + \frac{j-1}{a_i - 1}(g_i^* - g_{i*}) \quad \forall j = 1,2,\ldots,a_i \qquad \textbf{3.4.2-9}$$

The marginal value of an action is approximated by a linear interpolation, and thus, for every $g_i(a) \in [g_i^j - g_i^{j+1}]$ its utility will be given by equation 3.4.2-10.

$$u_i[g_i(a)] = u_i(g_i^j) + \frac{g_i(a) - g_i^j}{g_i^{j+1} - g_i^j}[u_i(g_i^{j+1}) - u_i(g_i^j)] \qquad \textbf{3.4.2-10}$$

The set of reference actions $A_R=\{a_1,a_2,\ldots,a_m\}$ is also "rearranged" in such a way that a_1 is the head of the ranking (best action) and a_m its tail (worst action). Since the ranking has the form of a weak order R, for each pair of consecutive actions (a_k, a_{k+1}) it holds either $a_k>a_{k+1}$ (preference) or $a_k \sim a_{k+1}$ (indifference). Thus, if

$$\Delta(a_k, a_{k+1}) = u'[g(a_k)] - u'[g(a_{k+1})] \qquad \textbf{3.4.2-11}$$

then one of the following holds:

$$\begin{cases} \Delta(a_k, a_{k+1}) \geq \delta \ \text{ iff } a_k \succ a_{k+1} \\ \Delta(a_k, a_{k+1}) = \delta \ \text{ iff } a_k \sim a_{k+1} \end{cases} \qquad \textbf{3.4.2-12}$$

where δ is a small positive number so as to discriminate significantly two successive equivalence classes of R. Taking into account the hypothesis on monotonicity of preferences, the marginal values must satisfy the set of the following constraints:

$$u_i(g_i^{j+1}) - u_i(g_i^j) \geq s_i \quad \forall j = 1,2,\ldots a_i - 1,\ i = 1,2,\ldots,n \qquad \textbf{3.4.2-13}$$

with $s_i \geq 0$ being indifference thresholds defined on each criterion g_i. Jacquet-Lagreze and Siskos (E. Jacquet-Lagreze and Y. Siskos 1982) urge that it is not necessary to use these thresholds in the UTA model but they can be useful in order to avoid phenomena such as $u_i(g_i^{j+1}) = u_i(g_i^j)$ when $g_i^{j+1} \succ g_i^j$.

The marginal value functions are estimated by means of the following Linear Program (LP) with 3.4.2-6, 3.4.2-7, 3.4.2-12, 3.4.2-13 as constraints and with an objective function depending on the and indicating the amount of total deviation:

$$\begin{aligned}
&[\min]F = \sum_{a\in A_R} \sigma(\alpha) \\
&\textit{subject to} \\
&\left.\begin{aligned} &\Delta(a_k, a_{k+1}) \geq \delta \ \text{ if } a_k \succ a_{k+1} \\ &\Delta(a_k, a_{k+1}) = 0 \ \text{ if } a_k \sim a_{k+1} \end{aligned}\right\} \forall k \\
&u_i(g_i^{j+1}) - u_i(g_i^j) \geq 0 \quad \forall i \text{ and } j \\
&\sum_{i=1}^{n} u_i(g_i^*) = 1 \\
&u_i(g_{i*}) = 0,\ \ u_i(g_i^j) = 0,\ \ \sigma(\alpha) \geq 0 \quad \forall \alpha \in \mathrm{A}_R,\ \forall i \text{ and } j
\end{aligned} \qquad \mathbf{3.4.2\text{-}14}$$

The stability analysis of the results provided by LP 3.4.2-14 is considered as a post-optimality analysis problem. As Jacquet-Lagreze and Siskos (E. Jacquet-Lagreze and Y. Siskos 1982) note, if the optimum *F*=0*, the polyhedron of admissible solutions for is not empty and many value functions lead to a perfect representation of the weak order R. Even when the optimal value F* is strictly positive, other solutions, less good for F, can improve other satisfactory criteria, like Kendall's τ.

As shown in **Figure 3.4-3,** the post-optimal solutions space is defined by the polyhedron:

$$\begin{aligned}
&F \leq F^* + k(F^*) \\
&\text{all the constraints of LP (3.4.13)}
\end{aligned} \qquad \mathbf{3.4.2\text{-}15}$$

where *k(F*)* is a positive threshold which is a small proportion of *F**. The algorithms which could be used to explore the polyhedron 3.4.2-15 are branch and bound methods, like reverse simplex method, or techniques dealing with the notion of the labyrinth in graph theory, such as Tarry's method. Jacquet-Lagreze and Siskos (E. Jacquet-Lagreze and Y. Siskos 1982), in the original UTA method, propose the partial exploration of polyhedron 3.4.2-15 by solving the following LPs:

$$\begin{aligned}
&[\min] u_i(g_i^*) \text{ and } [\max] u_i(g_i^*) \quad \forall i = 1, 2, \ldots, n \\
&\text{in polyedron (3.4.14)}
\end{aligned} \qquad (0.0.1)$$

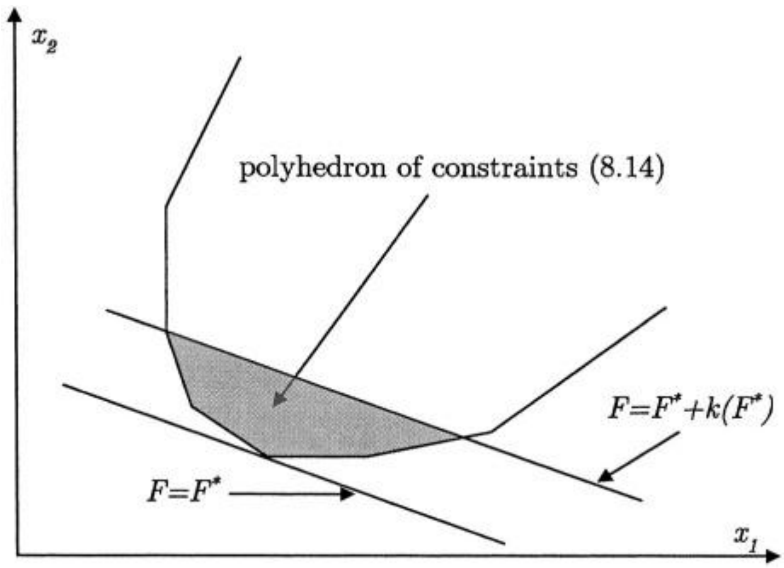


Figure 3.4-3: *Post-optimality analysis*

The average of the previous LPs may be considered as the final solution of the problem. In case of instability, a large variation of the provided solutions appears, and this average solution is less representative. In any case, the solutions of the above LPs give the internal variation of the weight of all criteria and consequently give an idea of the importance of these criteria in the DM's preference system.

One possible approach of applying MCDA in user modeling deals with the exploitation of MCDA techniques to model user preferences and store those preferences in a user profile.

In an early work of this thesis it was shown that different possible representations of preference information, stored in user profiles, lead to different results when these profiles are clustered. More specifically, it was shown that, when preferences are represented with utility information resulted from the application of the Disaggregation approach, as a candidate MCDA approach, significantly better results are attained in further processing of these user profiles, such as clustering. The comparison was made in regard to a second simpler preference representation, a binary representation of a ranking order on a set of alternatives. To maintain all the information provided by each user every ranking order was transformed into a binary vector. Since six products were ranked by every customer, a 30-tuple binary vector was formed for every user, containing their relative scores. An example is given below:

Suppose that user *i* provides a total ranking order of r_i = [2 2 1 3 5 4]. This means that he/ she prefers item 3 to item 1 and 2, between which there is an indifference relation. Consecutively, he/ she prefers item 4 to items 6 and 5 respectively. Let us assign 1 1 to an indifference relation, 1 0 to strict preference if the alternative's index is greater than its rank order and 0 1 in the reverse case. All possible combinations from the comparison of all alternatives are considered. This automatically produces the following assignment: [(2,2)→(1 1); (2,1)→(1 0); (2,3)→(0 1); (2,5)→(0 1); (2,4)→(0 1);…;(3,5)→(0 1); (3,4)→(0 1); (5,4)→(1 0)]. As a result, a ranking order like r_i would produce the binary vector b_i=[1 1 1 0 0 1 0 1 0 1 1 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1 0]. Therefore, the binary matrix to be used in the clustering procedure was a 204x30 dimension matrix coding all 204 users, in 204 binary vectors.

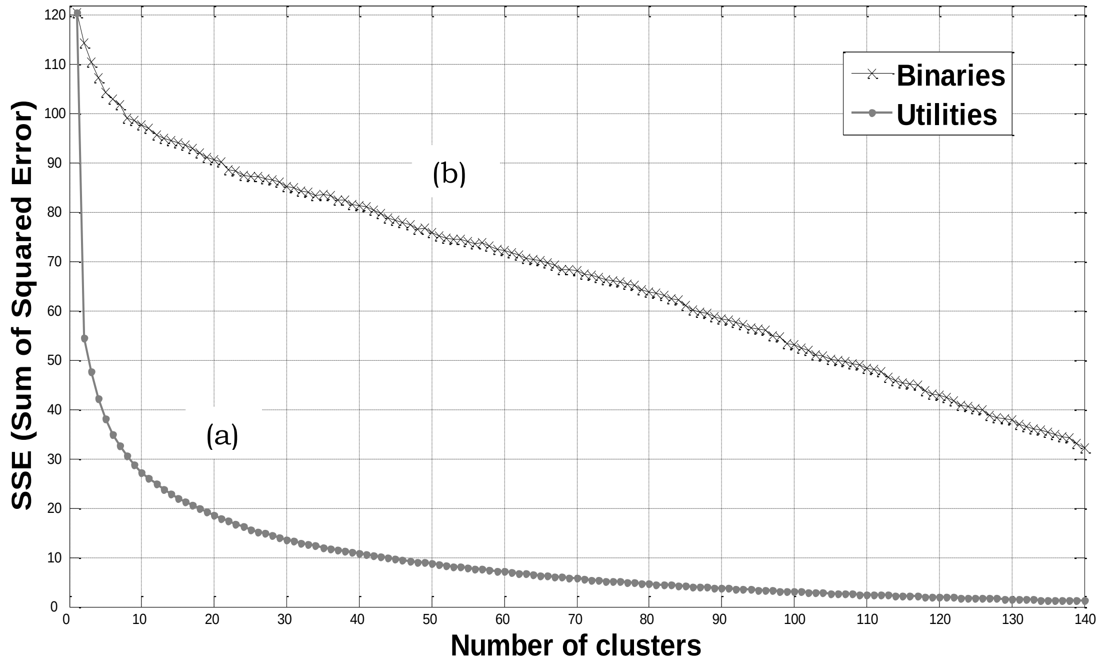


Figure 3.4-4: *Sum of Squared Error versus number of clusters a) "Utility matrix" results b) "Binary matrix" results*

In **Figure 3.4-4**, the Sum of Squared Error (SSE) is shown against the number of clusters formed by a clustering algorithm. The smaller SSE in (a) shows that data objects are closer to their correspondent centroids when information from utility functions is considered, which may in turn mean that MCDA preference information creates more accurate user profiles in terms of preference representation. Other kinds of representations of multiple criteria preference information are also possible, like the creation of user profiles based on the criteria weights as these are formed from the application of a UTA method. The latter is also the case of the proposed methodology presented in Chapter 4.

3.5 Conclusions

User profiling and modeling are both considered crucial steps in personalization technologies. The first, referring just to a way of representing user and user preferences or actions, is considered the first step of the user modeling process, a process based on various sophisticated techniques that aim at modeling user at different levels of cognition. The results of a user modeling process are often used to replace the user in several user-system interaction procedures.

A brief historical background on user modeling introduces this chapter, which further describes the attributes that together synthesize a complete user modeling process.

Five different dimensions under which a user model can be classified are given in section 3.3. This set is a first attempt of creating a taxonomy under which different user modeling methodologies can be classified.

In section 3.4 a Multiple Criteria Decision Analysis approach was presented, named the Disaggregation-Aggregation approach, as a potential methodology to serve user modeling issues. An implementation of this approach in the creation of user profiles that represent preference information based on the evaluation of alternatives on different criteria was also discussed and verified that MCDA methodologies can be proved useful in user profiling and modeling.

4 Methodological Framework

Contents

4.1 Introduction

In this chapter, the overall framework of the proposed approach together with the system's individual components is discussed. The methodological framework is based on four different consecutive phases, the *first phase*, during which input information is acquired and processed to attain an appropriate data input structure for the *second phase*, throughout which user modeling is employed. According to the discussed framework, a clustering process follows in the *third phase*, where the results of the second phase are clustered into meaningful, user profile groups. The methodological framework discussed in this chapter is integrated with the completion of the *fourth phase*, where the final recommendation is calculated. Furthermore, during this final phase, user feedback can also be received and processed accordingly.

At the final section, the UTARec system, a movie Recommender System that employs the discussed methodological framework is presented from a Graphical User Interface (GUI) perspective.

4.2 General framework

The proposed methodological framework is discussed in a general, yet complete approach, to ensure an application independent presentation.

Figure 4.2-1 summarizes the overall process structure, and the following subsections outline the steps involved. All the details are analytically discussed throughout this chapter, while the implementation of the proposed methodology via a the UTARec movie recommender system ensures a straightforward and complete presentation of the proposed framework. The proposed methodological frameworks is also discussed in (K. Lakiotaki, N. Matsatsinis, *et al.*).

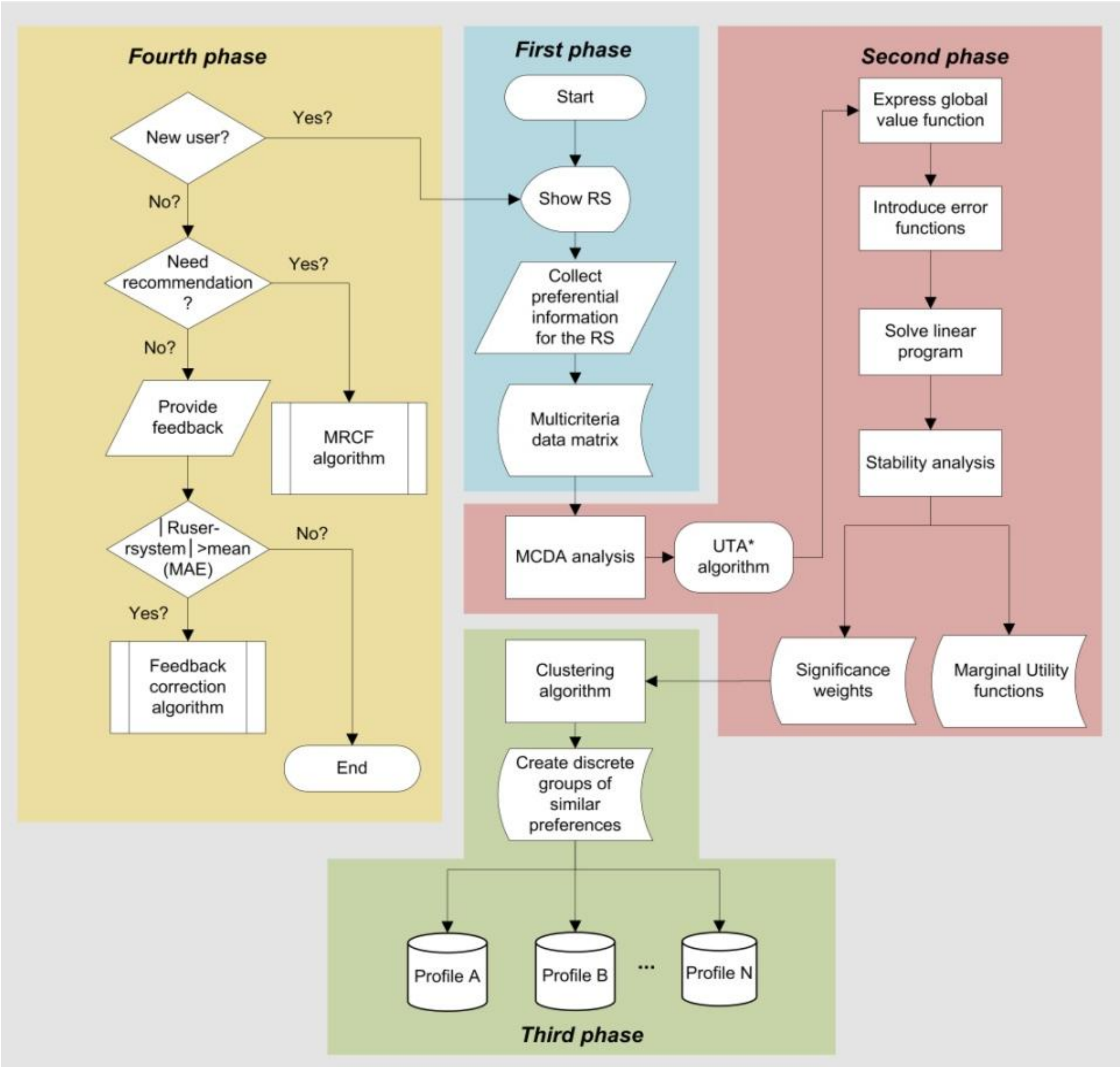


Figure 4.2-1: *Proposed system's build up architecture*

4.2.1 First phase: Data acquisition

Foremost to describing data acquisition procedure, it is essential to clarify at this point that two different types of data are gathered herein. Both types are attained by user statements. The first type concerns preference data given as numerical ratings, while the second type deals with preference statements in the form of a ranking order, the so called *weak preference order.*

Let A be the set of all alternatives (or more generally "possible worlds"). Then $\leq$ is a preference relation on A if it is a binary relation on A such that $a \leq b$

if and only if b is at least as preferable as a. It is conventional to say "*b is weakly preferred to a*", or just "*b is preferred to a*". If $a \leq b$ but not $b \leq a$, then the user strictly prefers b to a, which is written $a < b$. If $a \leq b$ and $b \leq a$ then the user is indifferent between a and b.

A *preference relation* underlies to the following assumptions:

- The relation is reflexive: $a \leq a$
- The relation is transitive: If $a \leq b$ and $b \leq c$ then $a \leq c$. Together with reflexivity this means it is a preorder
- The relation is complete: For all a and b in A, we have $a \leq b$ or $b \leq a$, or both (notice that completeness implies reflexivity). This means the user is able to form an opinion about the relative value of any pair of alternatives.
- The relation is continuous (if A is a topological space, then for each point x in A, the set of points that are strictly preferred to x and the set of points that x is strictly preferred to are both open).

If $\leq$ is both transitive and complete, then it is a *rational preference relation*. In some literature, also adapted herein, a transitive and complete relation is called a *weak preference order* (or total preorder).

To acquire user preference information, every user $u_t \in U$, where $t=1,2,\ldots,n$, n being the total number of users, is asked to evaluate a set of items $A_i \in A_R$, named the reference set A_R. For every alternative $A_i \in A_R$, $i=1,2,\ldots,m$, where m is the length of A_R, the user u provides a rating r_{ui}, for every criterion c_j, $j=1,2,\ldots,k$, where k is the total number of criteria, following a predefined measurement scale (i.e. 1 to 5). Additionally to these individual evaluations, the user is asked to provide an overall preference rank. These rankings of all alternatives that belong to the reference set are converted into a descending order to attain a *weak preference order*. Indifference relations are acceptable in the ranking order and are considered accordingly during the multi-criteria user modeling phase. With the completion of the data acquisition step, a data matrix is formed that acts as an input for the second phase. An example of such a matrix is show in **Table 5.3-1**.

4.2.2 Second phase: Multi-criteria user modeling

The multi-criteria input data matrix as acquired from the first phase, is analyzed and processed throughout the system's second phase, leading to the formation of a single k-dimensional vector for every user, referred as the *significance weight vector* or merely the *weight vector*. During the multi-criteria user modeling phase, the UTA* algorithm (Y. Siskos, E. Grigoroudis, *et al.* 2005), one of the most representative and widely applied Disaggregation-Aggregation framework algorithms is applied, to analyze user's cognitive decision policy. The UTA* algorithm, adopts the preference disaggregation principle, the philosophy of which is to assess/infer preference models from given preferential structures. More details on the Disaggregation-Aggregation approach can be found in Chapter 3 (3.4.2).

The first step of the Disaggregation-Aggregation approach deals with determining the problem statement in which the examining problem belongs to. Among the various problem statements that are met in Decision Aiding theory (D. Bouyssou, T. Marchant, *et al.* 2007), three problem statements are mostly appropriate in the case of a *recommendation problem*. These are: *choosing* one or more potential action/s from a set of actions (alternatives) *A*, *ranking* those alternatives in a descending order, or *sorting* them into predefined ordered categories. In general, there are various ways to present recommendations to the end user; either by offering the user the best item (choosing), or by presenting the top N items as a recommendation list (ranking), or by classifying the items into categories, i.e. "highly recommended", "fairly recommended", "not recommended" (sorting). Accordingly, a recommendation problem can equivalently belong to one of the first three problem statements, depending on its design architecture.

It is necessary to clarify at this point that although the UTA method that is performed in this step through the UTA* algorithm, belongs to the *ranking* problem statement, this does not imply that the recommendation problem should also belong to the same problem statement. To elucidate the inconsistency that seems to emerge at this point, it is simply mentioned that at the user modeling phase the problem to solve is to model user's value system and this is accomplished by means of the UTA method, however, the ultimate problem to solve is to predict ratings for unknown items.

Following the Disaggregation- Aggregation methodological schema (**Figure 3.4-1**), the modeling process of level 2 must conclude to a consistent family of criteria $\{g_1,g_2,\ldots,g_k\}$. More details on the criterion family requirements can be also found in (J. Figueira, S. Greco, *et al.* 2005). We briefly mention here that each criterion must be a non-decreasing, real valued function, defined on *A*, as follows:

$$g_j : A \to [g_{j*}, g_j^*] \subset \mathbb{R} / a \to \mathbf{g}(a) \in \mathbb{R} \qquad \textbf{4.2.2-1}$$

In 4.2.2-1, $[g_{j*}, g_j^*]$ is the criterion evaluation scale, g_{j*} and g_j^* are the worst and the best level of the j^{th} criterion respectively, $g_j(a)$ is the evaluation or performance of action *a* on the j^{th} criterion and $\mathbf{g}(a)$ is the vector of performances of action *a* on the *k* criteria.

The multi-criteria data input matrix is processed by the UTA* algorithm through an iterative ordinal regression procedure. Analytical details and an illustrative example of the UTA* algorithm can be found in (Y. Siskos, E. Grigoroudis, *et al.* 2005).

In abstract, the UTA* algorithm, considers as input a weak-order preference structure on a set of actions, together with the performances of the alternatives on all attributes, and returns as output a set of additive value functions based on multiple criteria, in such a way that the resulting structure would be as consistent as possible with the initial structure given by the user. This is accomplished by means of special linear programming techniques.

Four basic steps are followed in UTA* according to which, all the necessary parameters to estimate global value functions for each item and user are calculated.

The UTA* algorithm aims at estimating additive utilities of the form:

$$U(\mathbf{g}) = \sum_{i=1}^{m} u_i(g_i) \qquad \textbf{4.2.2-2}$$

subject to the following constrains:

$$\begin{aligned} & u_i(g_{i^*}) = 0 \quad \forall i \\ & \sum_{i=1}^{m} u_i(g_i^*) = u_1(g_1^*) + u_2(g_2^*) + \ldots + u_m(g_m^*) = 1 \end{aligned} \qquad \textbf{4.2.2-3}$$

where $u_i(g_i)$ $i=1,\ldots,m$ are non decreasing real valued functions, named marginal utility functions.

UTA* algorithm may be summarized in the following steps:

Step 1: Express the global value of reference actions $u[g(a_k)]$, $k = 1,2,\ldots,m$, first in terms of marginal values $u_i(g_i)$, and then in terms of variables w_{ij} according to the formula 4.2.2-4. The transformation of the global value of reference actions into weights values expression is made according to formula 4.2.2-5:

$$\begin{aligned} & w_{ij} = u_i(g_i^{j+1}) - u_i(g_i^j) \geq 0, \quad \forall i = 1,2,\ldots,n \\ & \text{and } j = 1,2,\ldots,a_i - 1 \end{aligned} \qquad \textbf{4.2.2-4}$$

$$\begin{cases} u_i(g_i^1) = 0 & \forall i = 1,2,\ldots,n \\ u_i(g_i^j) = \sum_{t=1}^{j-1} w_{it} & \forall i = 1,2,\ldots,n \text{ and } j = 2,3,\ldots,a_{i-1} \end{cases} \qquad \textbf{4.2.2-5}$$

Step 2: Introduce two error functions σ^+ and σ^- on A_{Ri} (reference set of alternatives) by writing for each pair of successive actions in the given ranking the formula 4.2.2-6:

$$\begin{aligned} \Delta(a_k, a_{k+1}) = {} & u[\mathbf{g}(a_k)] - \sigma^+(\alpha_K) + \sigma^-(\alpha_K) \\ & - u[\mathbf{g}(a_{K+1})] + \sigma^+(\alpha_{K+1}) - \sigma^-(\alpha_{K+1}) \end{aligned} \qquad \textbf{4.2.2-6}$$

Step 3: Solve the linear program (LP):

$$\begin{cases} [\min]z = \sum_{\kappa=1}^{\mu}[\sigma^+(\alpha_k)+\sigma^-(\alpha_k)] \\ \text{subject to} \\ \left.\begin{array}{ll} \Delta(a_k,a_{k+1}) \geq \delta & if\ a_k \succ a_{k+1} \\ \Delta(a_k,a_{k+1}) = 0 & if\ a_k \sim a_{k+1} \end{array}\right\} \forall k \\ \sum_{i=1}^{n}\sum_{j=1}^{a_i-1} w_{ij} = 1 \\ w_{ij} \geq 0,\ \sigma^+(\alpha_k) \geq 0,\ \sigma^-(\alpha_k) \geq 0\ \ \forall i, j \text{ and } k \end{cases} \quad \textbf{4.2.2-7}$$

Step 4 (stability analysis): Check the existence of multiple or near optimal solutions of the linear program 4.2.2-7. In case of non uniqueness, find the mean additive value function of those (near) optimal solutions which maximize the objective functions of 4.2.2-8, on the polyhedron of the constraints of the LP 4.2.2-7 bounded by the constraint of 4.2.2-9, where z^* is the optimal value of the LP in step 3 and ε a very small positive number.

$$u_i(g_i^*) = \sum_{j=1}^{a_i-1} w_{ij} \quad \forall i = 1,2,\ldots,n \qquad \textbf{4.2.2-8}$$

$$\sum_{k=1}^{m}[\sigma^+(a_k)+\sigma^-(a_k)] \leq z^* + \varepsilon \qquad \textbf{4.2.2-9}$$

By applying the UTA* algorithm all the necessary parameters to estimate global utility functions *U(**g**(a))* for each alternative are calculated. Thus, a value is assessed quantifying alternative's utility to each user and ensuring consistency with his/ her value system. UTA*'s output involves the value functions associated to each criterion, approximated by linear segments, as well as the criteria significance weights (trade-offs among the criteria values). The latter, expressed as a *weight vector* for every user, serves as his/ her value system information representation schema and provides the required user modeling data to proceed to the third phase, the clustering phase.

4.2.3 Third phase: Clustering

Generally, a clustering algorithm divides the original data set into disjointed groups. Clustering is an unsupervised process aiming at grouping data objects, based only on information found in the data that describes the objects and their relationships. The goal of a clustering algorithm is that the objects within a group should be similar (or related) as much as possible to one another, while they should be different from (or unrelated to) the objects in other groups. Most of existing clustering algorithms like the popular for its simplicity k-means (J. B. MacQueen 1967) are sensitive to initial parameters, such as the number of clusters and initial centroid positions. To limit these shortcomings, the global k-means (A. Likas, N. Vlassis*, et al.* 2003), a deterministic approach of the traditional k-means clustering algorithm, is enrolled in the third phase. Global k-means does not depend on any initial parameter values and employs the k-means algorithm as a local search procedure. Instead of randomly selecting initial values for all cluster centers, this algorithm acts in an incremental way, by optimally adding one new cluster centre at each stage, the one that minimizes a certain clustering criterion.

Suppose we are given a data set $\{x_1, x_2, \ldots x_n\}$, $x_n \in R^d$. The k-clustering problem aims at dividing this data set into k disjoint groups called clusters $C_1, C_2, \ldots C_k$, by optimizing a certain clustering criterion. The most widely used clustering criterion, adopted also in this case, is the Sum of Squared Error (SSE) between each data point x_i ($i=1,2,\ldots,n$) and the centroid m_j ($j=1,2,\ldots,k$) of the subset C_j which contains x_i. This clustering criterion depends on the cluster centers m_1, m_2, $..m_k$, and is shown in equation 4.2.3-1.

$$SSE(m_1, m_2, \ldots, m_K) = \sum_{i=1}^{N} \sum_{j=1}^{K} I(x_i \in C_j) \left| x_i - m_j \right|^2 \qquad \textbf{4.2.3-1}$$

Global k-means is applied to the set of user *weight vectors* and labels every user to a specific group. More formally the algorithm is shown in **Figure 4.2-2**.

Global k-means

```
1: Find the average of all n data vectors
k=2
2: repeat
3: Run k-means by considering the average
and i^th data vector as initial centroids
4: Keep the solution with the minimum
clustering error
5: Update centroids
k=2+i
6: repeat
7: Run k-means by considering the centroids
obtained from 5 and the i^th data vectors as
initial centroids
8. until convergence
```

Figure 4.2-2: *Pseudo-code of the Global k-means algorithm*

Although Global k-means does not provide any evidence for the optimal number of clusters, which is indeed considered an ambiguous and highly application dependent process, it ensures optimality at any step. This means that by applying this algorithm it is assured that an optimal solution is attained for any number of clusters we may further decide that is appropriate for the specific problem to solve.

4.2.4 Fourth step: Recommendation phase

Following the formation of user groups with similar preferences (user profile clusters), accurate item recommendations can be provided to these users. The recommendation phase is accomplished by implementing the collaborative filtering philosophy inside each user group.

The *multidimensional Multi-criteria Collaborative Filtering (MRCF-dim)* approach applied herein, is based on multidimensional distance metrics. First, it calculates the distance between two users, *u* and *u'*, for the same item, according to equation 4.2.4-1:

$$d_{uu'} = \sqrt{\sum_{n=1}^{k+1} r_{un} - r_{u'n}}^{\;2} \qquad \textbf{4.2.4-1}$$

In 4.2.4-1, r_u is the rating vector of user u and $r_{u'}$ the rating vector of user u'. By rating vector we mean the set of ratings that user u provided for an item i, including the overall rating and $k+1$ represents the dimension of this rating vector, as a result of the k criteria and the overall rating that altogether define user vector's dimensionality.

Second, the overall distance between two users u and u' is simply given by equation 4.2.4-2.

$$dist(u,u') = \frac{1}{|U(u,u')|} \sum_{i \in U(u,u')} d_{uu'} \qquad \textbf{4.2.4-2}$$

In 4.2.4-2, $U(u,u')$ denotes the set of items that both u and u' have rated. This means that the overall distance between two users, u and u', is the average distance between their ratings for all their common items.

Finally, users' similarity, which is inversely related to their distance, is given by:

$$sim(u,u') = \frac{1}{1 + dist(u,u')} \qquad \textbf{4.2.4-3}$$

This notion of similarity ensures that the similarity will approach 0 as the distance between two users becomes larger, and it will be 1 if two users rated all their common items evenly (G. Adomavicius and Y. Kwon 2007).

Note that $sim(u,u')$, is calculated if and only if, u' belongs to the same group with u and their $U(u,u')$ is not empty (henceforth, we will refer to these users as "mates"). Therefore, the computational effort is minimized compared to traditional non clustering approaches that compute $sim(u,u')$ for all possible user combinations.

After calculating a similarity index for "mate" users, equation 4.2.4-4 provides a potential rating $R(u,i)$ for any unexplored item i.

$$R(u,i) = \left(\frac{1}{\sum_{u' \in C(u)} sim(u,u')} \right) \cdot \sum_{u' \in C(u)} sim(u,u') \cdot R(u',i) \qquad \textbf{4.2.4-4}$$

Equation 4.2.4-4 is in fact a similarity weighted sum of known ratings and *C(u)* defines user's neighborhood, meaning the cluster where u belongs to.

Depending on the data set's magnitude, or on the "popularity" of the item to be recommended, the space *C(u)* may be empty. Should this be the case, the coefficient of similarity changes to include a greater space, the next closest to *C(u)* cluster. Thus the new similarity coefficient becomes:

$$sim\_new = sim\left(C(u),C(u')\right) * sim(u,u') \qquad \textbf{4.2.4-5}$$

Where

$$sim\left(C(u),C(u')\right) = 1/\left(1 + dist\left(C(u),C(u')\right)\right) \qquad \textbf{4.2.4-6}$$

Is the distance between cluster centers *C(u)* that user u belongs to and *C(u')* that user *u'* belongs to. Thus, equation 4.2.4-4 becomes:

$$R(u,i) = \left(\frac{1}{\sum_{u' \in C(u)} sim\_new(u,u')} \right) \cdot \sum_{u' \in C(u)} sim\_new(u,u') \cdot R(u',i) \qquad \textbf{4.2.4-7}$$

The algorithm that the recommendation phase follows is shown in **Figure 4.2-3**.

Recommendation phase

```
Find all users that have rated i and belong to C(u)
if C(u) is empty
repeat
Find closest to C(u) cluster C' by minimum cluster center distance
Apply equation 4.2.4-4
until non empty C'
```

Figure 4.2-3: *Pseudo-code of the recommendation algorithm*

In the event that no user has ever rated the specific item on a range of 10% of the total number of clusters formed, a problem commonly mentioned in the Recommender Systems literature as the Cold Start or First Raster Problem, the system provides a recommendation based on the average user recommendations of cluster *C(u).*

4.2.5 Feedback mechanism

System's feedback mechanism is activated by a user, when he/she is willing to provide a rating for an item that explored, according to system's preceding recommendations. In the case where a user disagrees with the recommendation given and provides the rating that he/she would give to the specific item, the system processes this information, by triggering the feedback correction algorithm. According to this algorithm, the new user value $R_{actual}(u)$ is compared to past system value $R_{system}(u)$ in terms of absolute difference. If this difference is greater than the mean absolute difference stored for this user MAE_u, then this alternative is included in the reference set and the UTA* algorithm runs again to calculate a new *significance weight vector* for this user. This *weight vector* will indicate whether this particular user should belong to a different group. To decide this, the feedback correction algorithm calculates the Squared Euclidean distance (SE) of the user's weight vector from every centroid of the formed groups. This particular user will now belong to the group where user's SE is less or equal to group's maximum SE. According to the results of the feedback function, the system updates, or not, the specific user profile, which in other words means, that may change or not the group that this user belongs to.

A general pseudocode scheme of feedback algorithm is shown in **Figure 4.2-4:**

Feedback algorithm

```
1: Find dif=abs(R_actual(u)-R_system(u))
2: if diff is greater or equal to MAE_u
3: run UTA*
4: Calculate new significance weight w_new
5. Find closest to w_new cluster C by minimum
cluster center distance
6: Assign user to closest cluster
```

Figure 4.2-4: *Pseudo-code of the feedback algorithm*

4.3 UTARec System

The proposed methodology described analytically hitherto, is demonstrated via a movie Recommender System implementation described in this section. The Multiple Criteria Movie Recommender Systems is called UTARec, since it is a **UTA** based **Re**commender System. The system has been developed as a windows application in MATLAB version 7.6.0.324 (R2008a).

UTARec's front page can be described as a simple log in page, where existing users either log in to the system or new users register to the system.

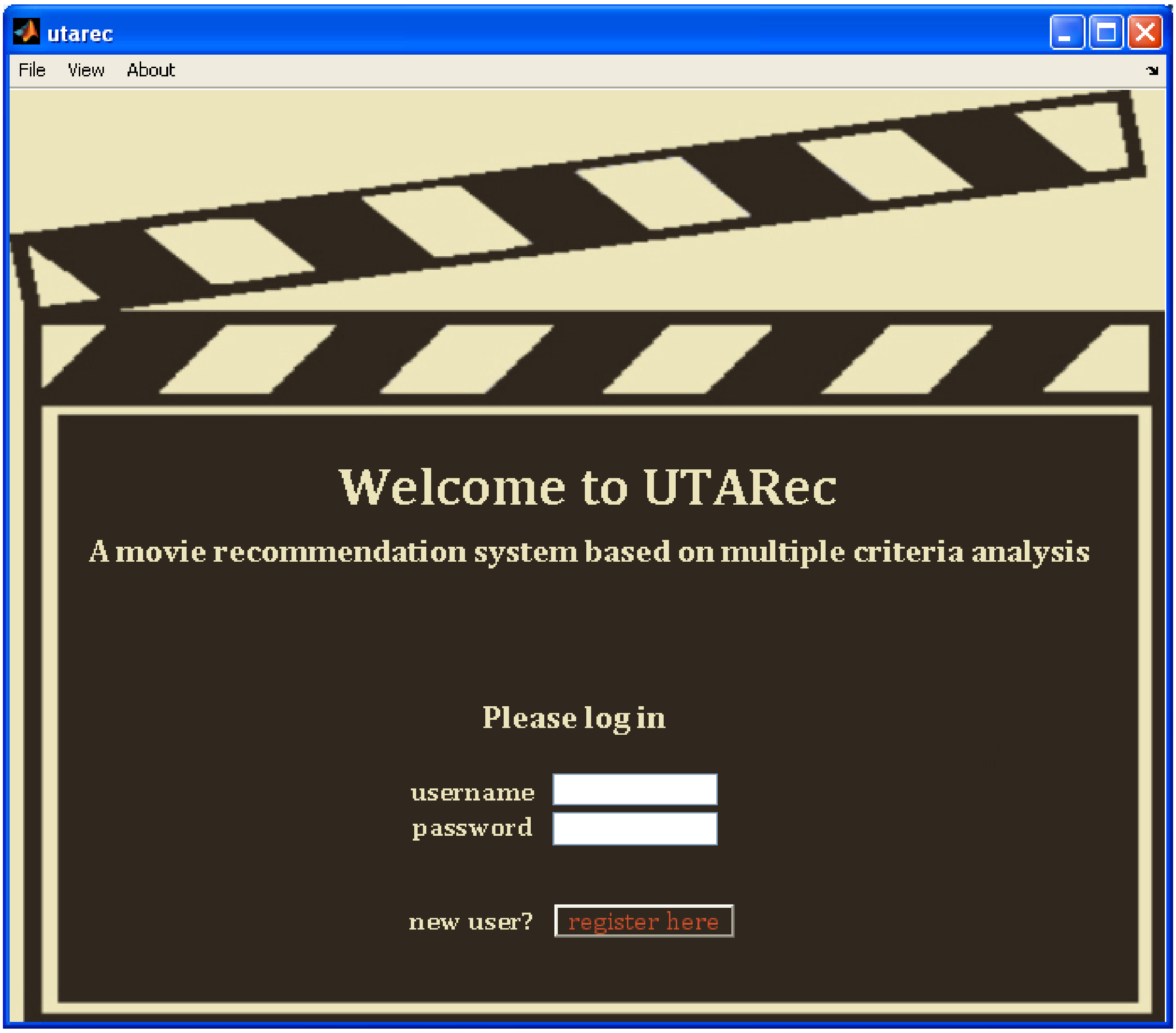


Figure 4.3-1: *UTARec's log in page*

System's menu capabilities are shown in **Figure 4.3-2.**

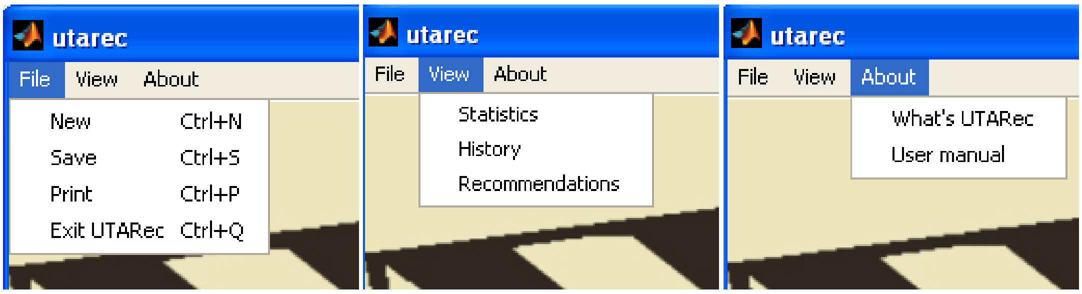


Figure 4.3-2: *UTARec's menu capabilities*

In UTARec a user by selecting **file**, he/ she may open a new window and start a **new** session, may **save** current sessions results, **print** them or **exit** the application.

By selecting **view** from the menu bar, the user may access his/ her statistics information, for instance, how many movies he/ she has rated, view the history, i.e. see a list of all movies that has been recommended in the past, or just navigate to the UTARec's recommendation page.

The **about** menu group provides information on UTARec methodology as well as information on its publications, while the user manual contains instructions on how to interact with the system.

In case of a new user, UTARec directs the candidate user to the registration page, where initial preference information is elicited, to create the alternatives' reference set as analytically mentioned in 4.2.1. The registration page is shown below in **Figure 4.3-3**. As soon as the user completes the registration form, UTARec enables phases 2, 3 and 4 of the proposed methodology and thus calculates a recommendation rate. This rate corresponds to the potential rate that this specific user would give when he/ she watches any movie. This rate is calculated for all movies that "mate" users (users that belong to the same group) have rated.

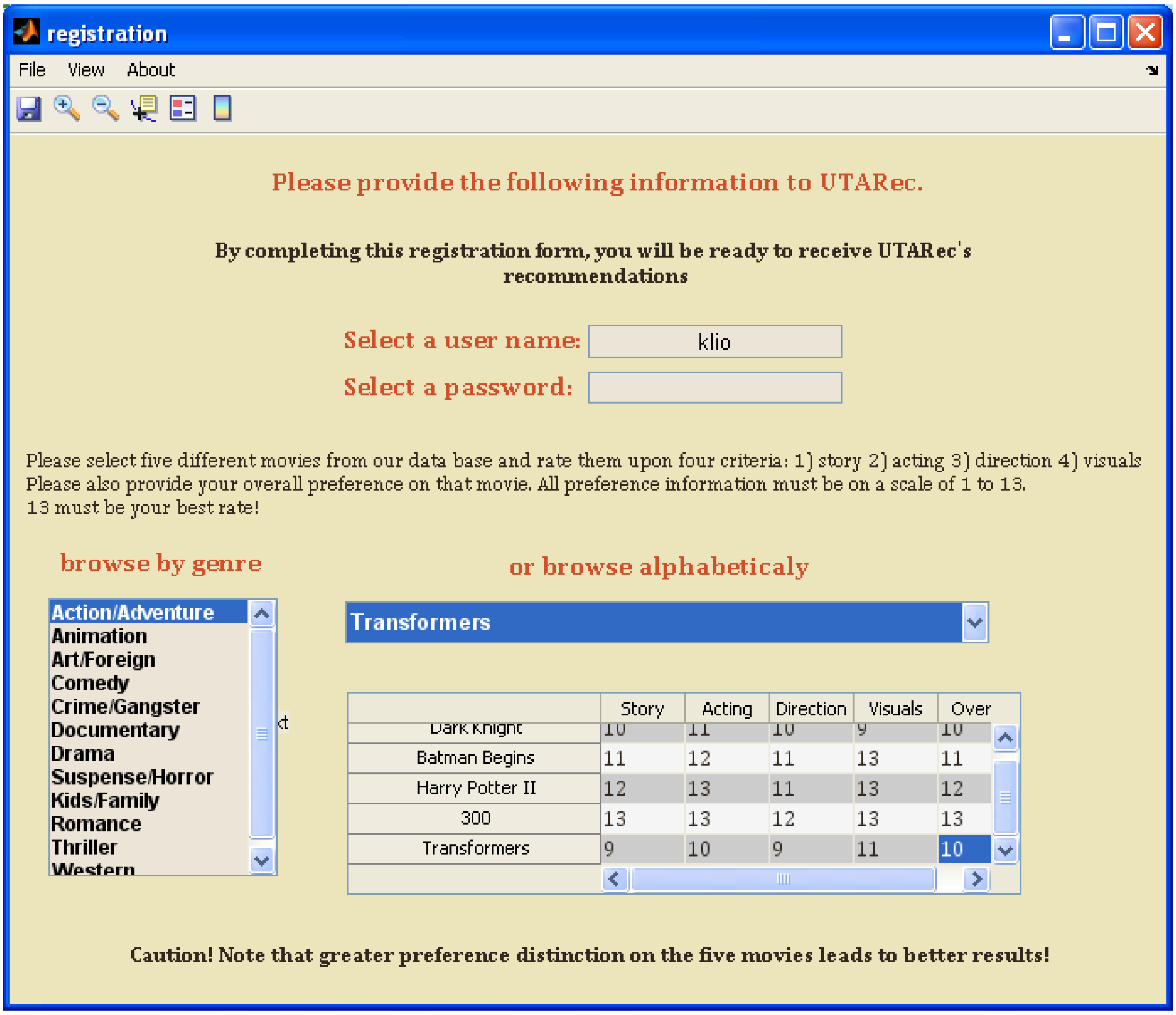


Figure 4.3-3: *UTARec's registration page*

Suppose that an existing user has logged in, he/ she is directed on the welcome page (see **Figure 4.3-4**) where a decision on whether he/ she is looking for a recommendation on a specific movie that he/ she is ready to watch, or need recommendations for any movie that system may recommend.

Figure 4.3-4: *UTARec's user welcome page*

If the particular user needs recommendations for a movie that has already decided to watch, the page shown in **Figure 4.3-5** appears, with UTARec's predicted value on this particular movie and also some useful statistical information on that specific movie.

The button on the left, "*see UTARec's recommendations*", directs the user to the UTARec recommendation page, where system provides recommendations on all movies that user's "mates" have rated and the interacting user has not watched yet. The user may browse these movies either alphabetically or by genre. **Figure 4.3-6** shows a screenshot of that page for user "*klio*".

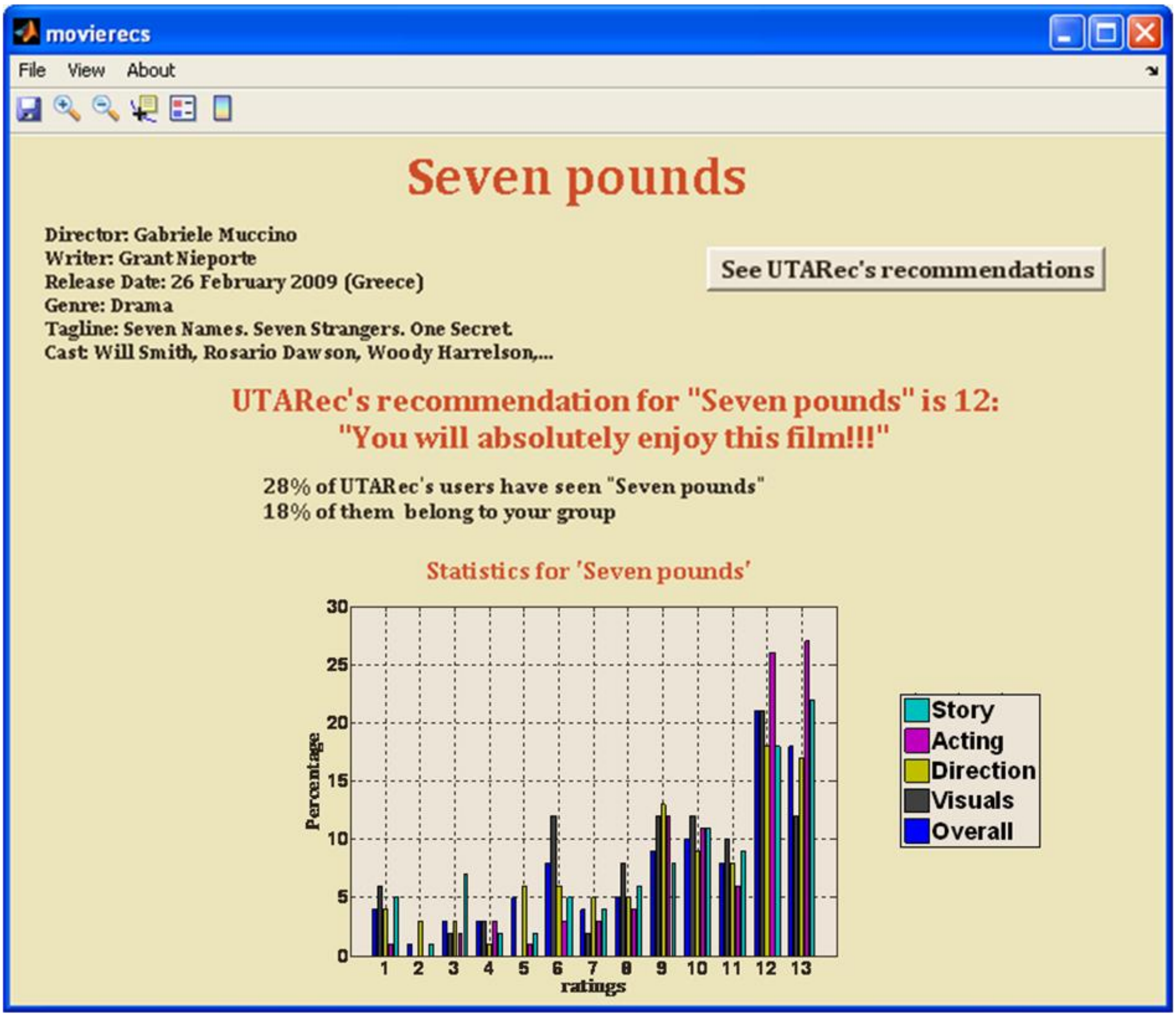


Figure 4.3-5: *Recommendations on a predetermined by the user movie*

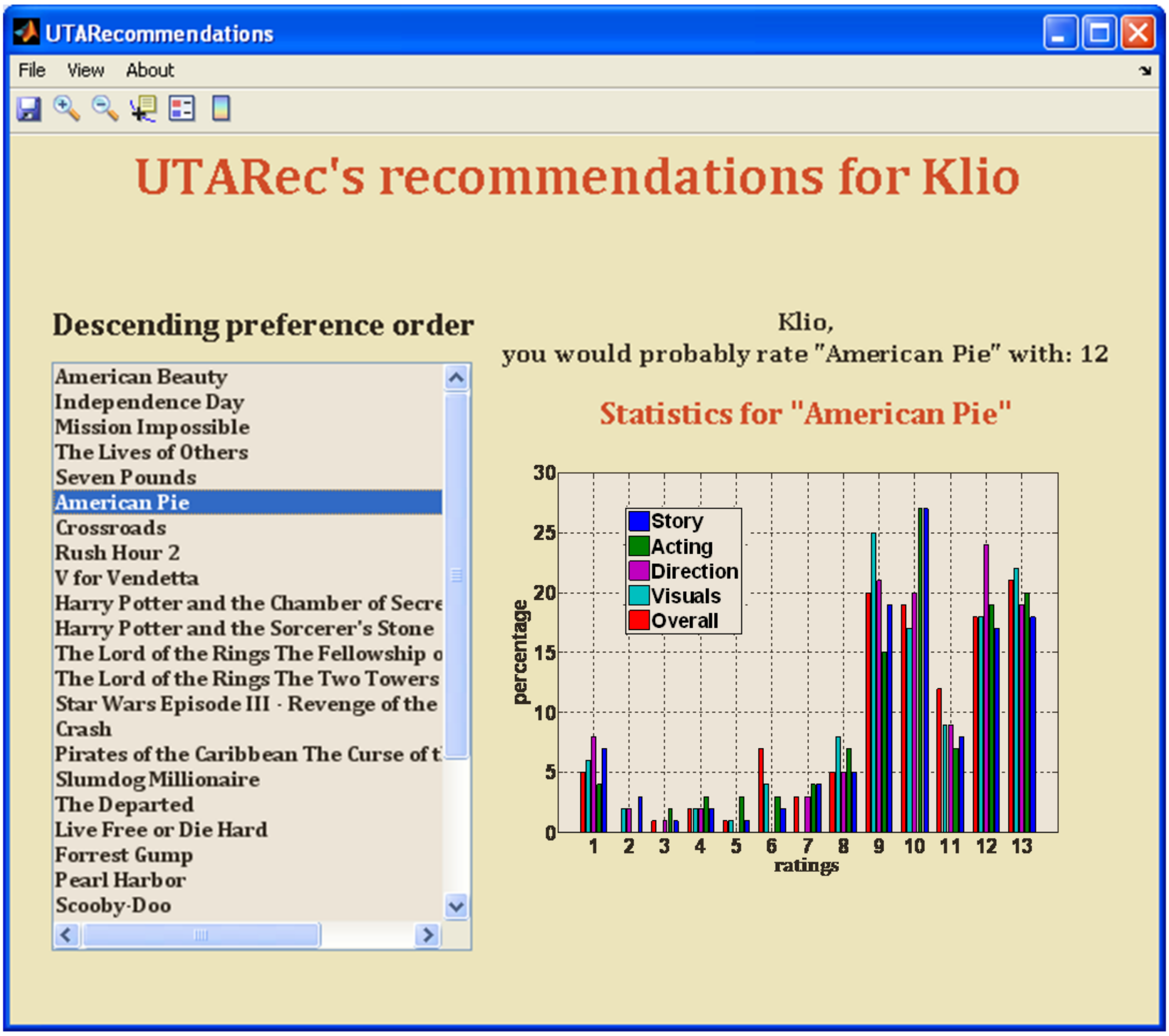


Figure 4.3-6: *UTARec's recommendations for a particular user*

4.4 Conclusions

In this chapter the individual components that when composed together construct the overall methodological framework of the proposed approach, were presented in a consecutive phase mode.

Without loss of generality, all the specific steps undertaken to build a system based on the proposed approach were discussed. The multiple criteria nature of the proposed methodology is attributed to the UTA* algorithm that is incorporated into the user modeling phase. Subsequently, a deterministic approach of the k-means algorithm is employed to cluster user profiles, the output of the user modeling phase, into user groups of similar value system. The hybrid aspect of the proposed methodology is due to the collaborative filtering philosophy that is encompassed to provide recommendations.

The proposed methodological framework is demonstrated by the UTARec system that was also presented in section 4.3. UTARec is an integrated multi-criteria movie Recommender System, which provides recommendations to candidate users as soon as they log in to the system.

5 Results

Contents

5.1 Introduction

UTARec is evaluated in this chapter under multiple points of view, to robustly support its performance. Initially, focus is given on the evaluation of system's recommendation accuracy, while subsequent evaluation analysis, concerns system's overall functionality, where additional parameters, like the reference set size or the personalization flexibility are considered.

Chapter 5 begins with a brief referance to a preliminary version of UTARec, a purely Multiple Criteria Decision Analysis (MCDA) application, together with its results. These results revealed the necessity of incorporating additional aspects to increase system's functionality. These aspects fall into the area of traditional Recommender Systems methodologies, such as the collaborative filtering method. UTARec's preliminary studies are mentioned to demonstrate the inevitability of combining technologies from both fields, the MCDA and the Recommender Systems.

In the remainder of this chapter the performance of the latest and complete version of UTARec is discussed.

5.2 Preliminary results

A preliminary version of UTARec was published in the 2nd International Conference of Recommender System that was held in Lausanne on October 2008 (K. Lakiotaki, S. Tsafarakis, *et al.* 2008). In that work, a first attempt to demonstrate the potential of applying MCDA methodologies was implemented. However, this was a pure application of the Disaggregation-Aggregation approach as a movie Recommender System. In other words, only the Disaggregation-Aggregation approach represented by the UTA* algorithm, in the user modeling phase was maintained to the final version of UTARec. This first attempt of incorporating MCDA in Recommender Systems mainly focused on verifying the prediction accuracy of the Disaggregation-Aggregation approach.

The data set used in the specific experiments included 201 users and a total of 2,694 multiple criteria ratings. The number of movies that each user had rated varied along the users and from 7 to 25. The collected data came from 45 randomly selected movies encoded with a serial number from 1 to 45. This data set was of the same form as the three data sets used in the final version and are analytically described in the subsequent section (5.3).

The Disaggregation-Aggregation approach as described in 3.4.2 was applied to the data set and the marginal utility functions were calculated for each user based on his/her preference data on a reference set of five alternatives, movies in this specific application. In **Figure 5.2-1** the marginal utility functions are displayed for all four criteria upon which every user had rated the alternatives reference set A_R. These functions were used to calculate a utility score for every alternative excluded from the reference set according to equation 4.2.8 of chapter 4.

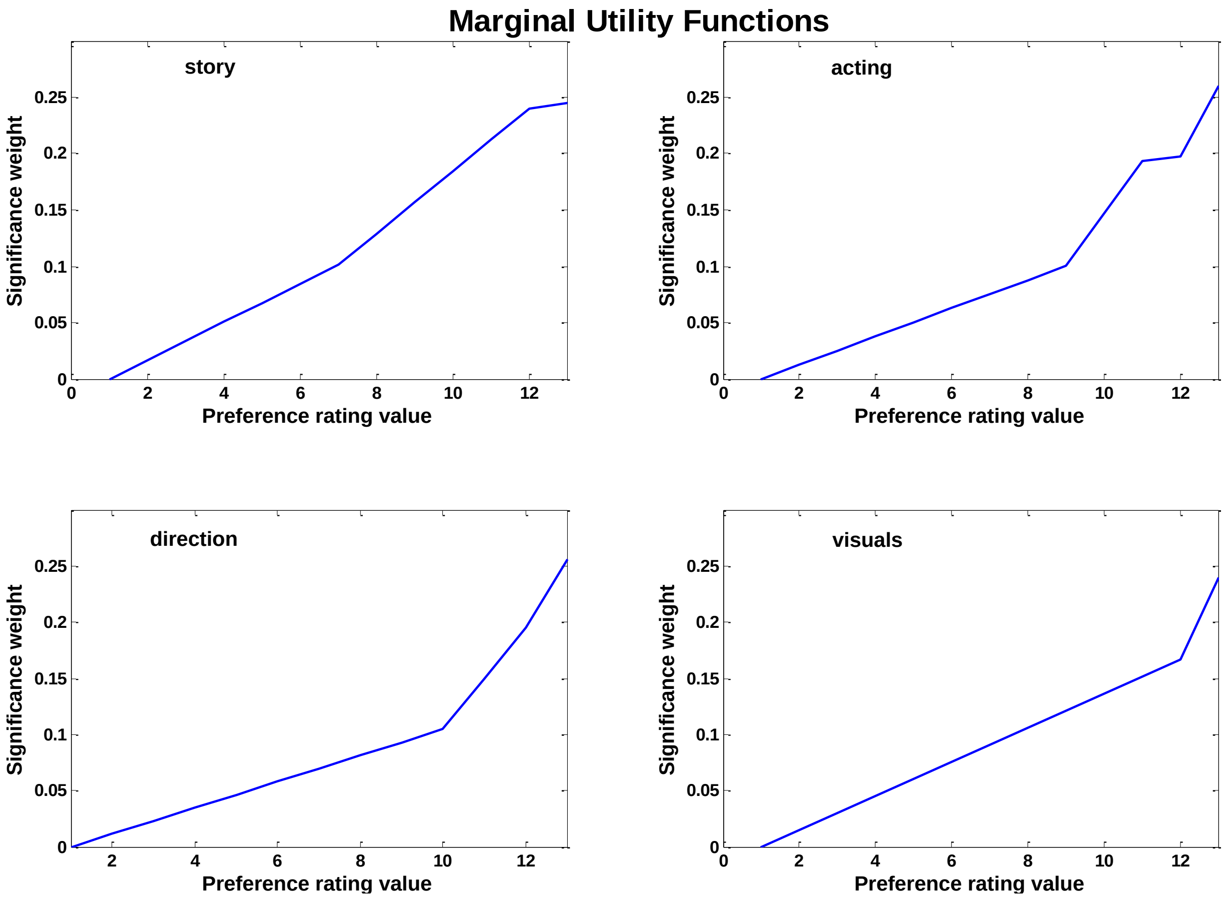


Figure 5.2-1: *Criteria Marginal Utility Functions for a characteristic user*

To evaluate the recommendation performance of UTARec system, first, the prediction performance was calculated. Specifically, Kendall's tau was calculated between user actual ranking order and system's predicted ranking order. In addition, to further verify the results, ROC curve analysis was implemented to reveal system's classification accuracy. For the later to be applied, a grouping of the actual user answers was preceded. As already stated, each user had rated a different number of movies ranging from 7 to 25. The prediction accuracy of UTARec was calculated on the residual set of movies (test set) after the exclusion of the reference set (training set) used in UTA*. For the test set any information upon the overall rating by the user was ignored and calculated by the system by linearly combining the marginal utility functions and the actual criteria ratings.

Since the UTA* algorithm employs ordinal regression techniques, an appropriate metric to evaluate the results of such an algorithm is Kendall's tau, a measure of correlation between two ordinal-level variables (see also 2.6.3). In this case the first variable is the ranking order of the test set alternatives as stated by the user and the second is the ranking order of the same alternatives depicted from the total utility values calculated by the model. Kendall's tau varies between -1 and 1 with 1 indicating a total agreement of the orders. **Figure 5.2-2** shows values of Kendall's tau for each user. A mean value of 0.74 was found with 20,4% of users having a Kendall's tau of 1. This result indicates that the two ranking orders compared, the one predicted by UTARec and the actual one as stated by the user, are in high agreement.

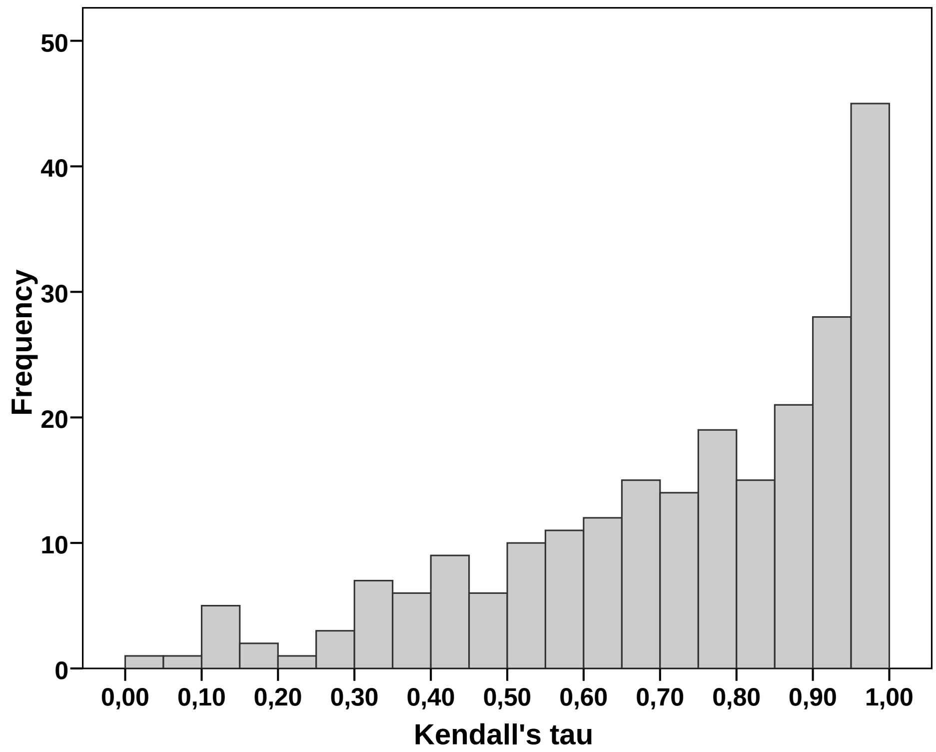


Figure 5.2-2: *Kendall's tau between user's and UTARec's ranking order per user*

An ROC graph depicts relative trade-offs between true positives and false positives. The diagonal line of an ROC graph represents the case of randomly guessing a class. Furthermore, the Area Under the Curve (AUC) has been shown to be an accurate evaluation measure and is widely used in applications where ranking is crucial (see more details in 2.6.2). In **Figure 5.2-3** the ROC graph for the results of the preliminary UTARec is depicted. The Area Under Curve was found to be 0.81 for 50 cut off points.

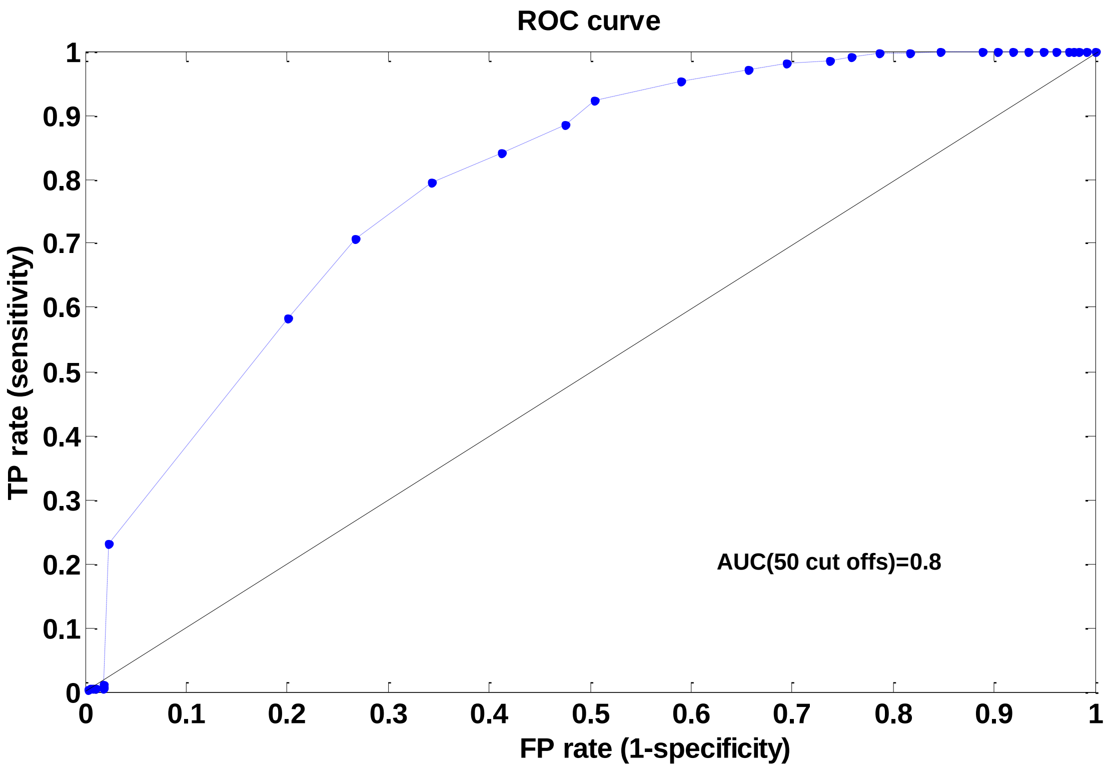


Figure 5.2-3: *Receiver Operating Characteristic curve for 50 cut off points*

The high prediction accuracy of this first implementation indicated that UTARec may act as a promising candidate for Recommender Systems. To this end, further parameters to improve UTARec's functionality were studied and resulted to the latest version of UTARec as presented in this thesis. The major drawback of applying MCDA methodologies and more specifically the UTA* algorithm to design a Recommender System lies in the relatively vast amount of data that the user must provide to the system, increasing thus not only user effort but also systems utility. In the specific preliminary application for example, preference information on individual criteria must be a priori known in order for the system to calculate the overall utility for a movie. This makes

UTARec a useless, in terms of user functionality, tool since a user must first watch the movie to be able to provide this information. This drawback is met in any system that handles qualitative criteria. Nevertheless, if only quantitative criteria are used to model user decision policy, a Multiple Criteria Recommender System can be built simply as described in (K. Lakiotaki, S. Tsafarakis, *et al.* 2008).

The demand of a Recommender System to process quantitative criteria, led to the conception of the methodological framework presented in chapter 4 of this thesis. The predecessor of the ultimate UTARec, as discussed thus far, acted as a basis and inspired its further development, leading to the foundation of the latest and complete version of UTARec.

5.3 Data sets description

In the commercial sector, Yahoo!Movies has launched a recommendation service that employs user-specific multi-criteria ratings for different movies (movies.yahoo.com). The initial experimental data set was retrieved from Yahoo!Movies, where users provided preference information on movies based on four different criteria. The four attributes that constituted the criteria family were: c_1=*acting*, c_2=*story*, c_3=*direction* and c_4=*visuals*. All values were measured in a 13-fold qualitative scale with F denoting the worst evaluation grade and A+ declaring the most preferred value. For processing purposes, letters were replaced with numbers, so as 1 corresponded to the worst value, formerly denoted as F and 13 to the best value, A+. In addition to individual criteria ratings, users were asked to provide an overall grade, which reflected their global preference over each movie. An example of the initial and the final, after the transformations rating scheme, is shown in **Table 5.3-1**. The left side shows a typical raw data form, while in the right side the same data is presented in a ready to process form (final data form).

Initial data form						Final data form					
user_id	*Overall grade*	*C_1*	*C_2*	*C_3*	*C_4*	*movie_id*	*user_id*	*Ranking order*	*C_1*	*C_2*	*C_3*	*C_4*	*movie_id*
1	*A+*	*A*	*A*	*A*	*A-*	1	1	*1*	*1*	*1*	*1*	*1*	1
	B+	*B*	*A*	*B*	*A*	4		*2*	*1*	*1*	*9*	*1*	4
	B	*B*	*A-*	*B*	*A*	25		*3*	*9*	*1*	*9*	*1*	25
	B-	*B*	*B*	*B*	*B*	23		*4*	*1*	*1*	*9*	*9*	23
	C+	*C*	*B*	*C*	*A*	9		*5*	*6*	*9*	*7*	*1*	9
2	*A*	*A*	*A-*	*A-*	*A*	9	2	*1*	*1*	*1*	*1*	*1*	9
	B+	*B*	*B*	*B*	*B*	18		*2*	*1*	*9*	*9*	*9*	18
	B+	*A-*	*A-*	*A*	*B*	2		*2*	*1*	*1*	*1*	*9*	2
	…	…	…	…	…	…		…	…	…	…	…	…

Table 5.3-1*:* *A sample of the multi-criteria data input matrix before (left side) and after (right side) preparation.*

Data cleaning followed soon after the data acquisition phase to remove any case with at least one or more not available values. This introduced 18% shrinkage in our data set. A subsequent filter to cut off users with less than five rated movies was applied, to assure an adequate set of evaluated movies for every user. The data cleaning workflow process is show in **Figure 5.3-1.**

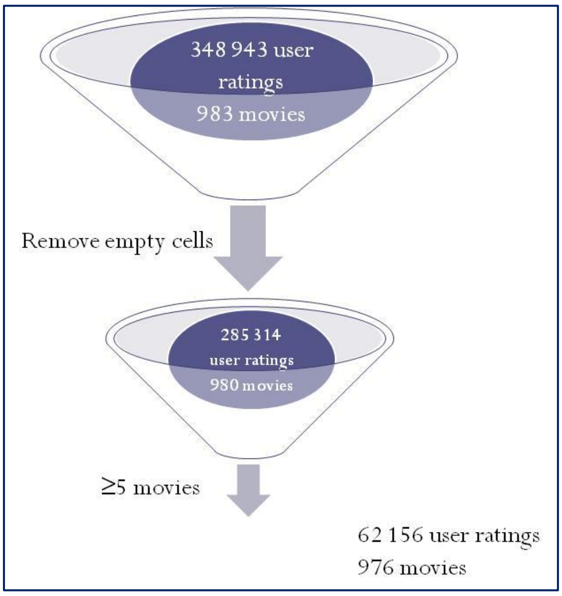


Figure 5.3-1: *Workflow diagram of data cleaning*

Three different data sets were prepared for the three different evaluation schemes. For the first scheme where recommendation accuracy is primarily examined, the entire set of users that had rated at least 5 movies was used, while the rest two data sets are used for the overall functionality.

The second and third data sets, both are filtered out from the initial, with additional rules applied to them. For the second data set the following restrictions are considered:

1) All users have rated **at least 10** movies

2) **Five** of them were used to model user preferences (training set)

3) The remaining movie set (test set) was used solely for evaluation purposes

The difference of the first and the second data set, is the fact that in the second, a minimum number of five items is evaluated, which ensures that for all users, the number of items used for evaluation purposes is at least equal or greater than that used for modeling. On the one hand this ensures a sufficient evaluation set, notably smaller on the other.

The third data set, the smallest of all, was used to study how the reference set size affects the recommendation accuracy of the proposed system. Again, filtered out from the initial one, this latter data set included all users that had rated at least 35 movies. This decision was taken to ensure that at least 5 movies would remain for evaluation, while the reference set was incrementally increased from 5 to 30 items, with steps at 10 and 20 items. The corresponding distributions of the three data sets used in the evaluation analysis are provided hereupon.

A schematic representation of the preparation of the three data sets is shown in **Figure 5.3-2** where it is obvious that the first set acts as a superset of the two other data sets.

Figure 5.3-2: *Preparation of the three data sets.*

It is crucial to mention at this point, that although Recommender Systems' research area is used to very large data sets, i.e. the Netflix prize dataset consists of about 480 thousand users, these datasets consist of an overall single rating (usually in a scale of 1 to five stars) and do not provide any information on individual criteria. Generally, it is not very easy to come across data sets with preference information on several attributes, since it is commonly believed that people are unwilling to provide a lot of information. It is advocated in this thesis that preference information on individual criteria offer valuable knowledge for the design and effectiveness of Recommender Systems, as it can be processed to build user's value system and decision policy, so, asking the user to provide this information leads to significant improvement of recommendation accuracy, which can anticipate user's additional effort.

5.3.1 First data set description and statistics

For the first evaluation scheme, the resulting experimental data set included 6078 different users that rated at least 5 different movies and 976 movies in total. The overall number of ratings was 62156 and every user has rated about 10 movies on average. The maximum number of movies that a user had rated was 237. The average evaluation grade was 9.6, 9.9, 9.5, 10.5 and 9.6 for the criteria *acting, story*, *direction*, *visuals* and *overall*, respectively.

In **Figure 5.3-3** a histogram of the ratings that the users provided is shown for all criteria as well as for the overall preference. Obviously, high rating values seem to be favored by movie raters.

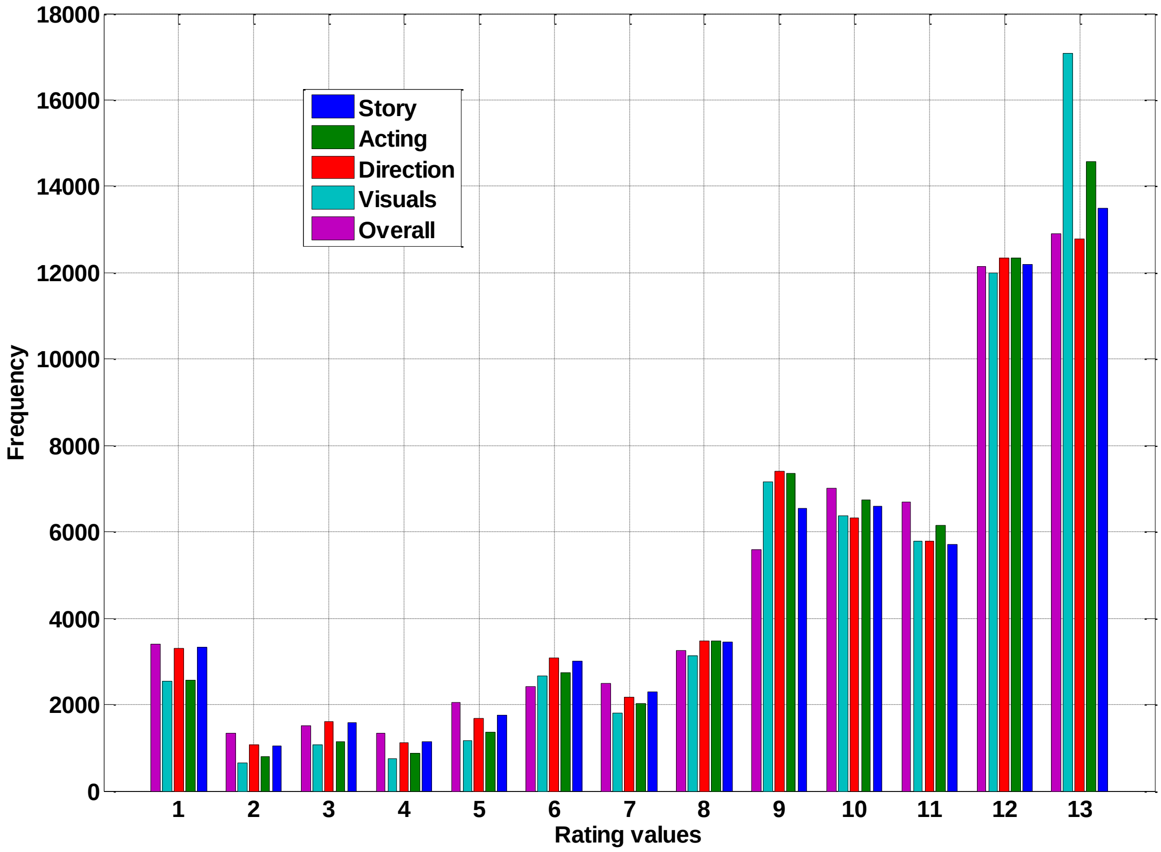


Figure 5.3-3: *Distribution of the first data set. The five criteria and the overall statements correspond to the six different color bars as indicated in the legend for RS>5.*

To obtain an idea of the frequency by which movies had been rated, the frequency of rated movies is provided in **Figure 5.3-4**, classified into 6 different groups. Even though there is no "optimal" number of bins to define in a histogram, the creation of six groups seems a rational choice in this case. The first group (<10) consists of the number of movies that have been rated no more than 10 times. These movies are referred as the “not popular” rated movies. The second class consists of movies that have been rated more than 10 and less than 50 times. This seems to be the highest populated class, since 409 (42%) movies belong to this group. Movies that belong to the last group (>500) have been rated at least 500 times, which makes them the “most popular” movies on this dataset. The frequency rating ranges between 1 and 830 and the average rating for a movie is about 64 times.

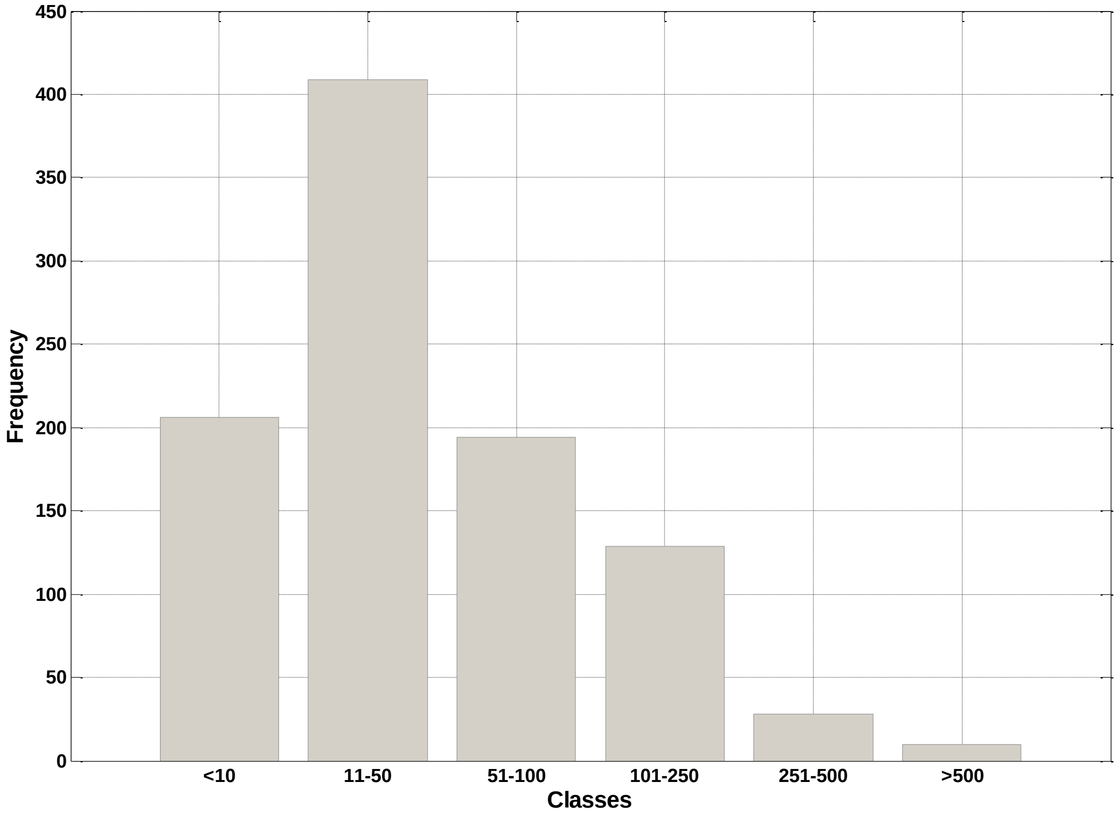


Figure 5.3-4: *Frequency of rated movies per class for the first data set*

However, in the distribution of the number of rated movies by every user, it is noticed that about 28% of users had rated 5 movies and only about 11% of the users had rated over 15 movies as also shown in **Figure 5.3-5**. The number of rated movies by users exhibits an exponential decrease as depicted in **Figure 5.3-5**.

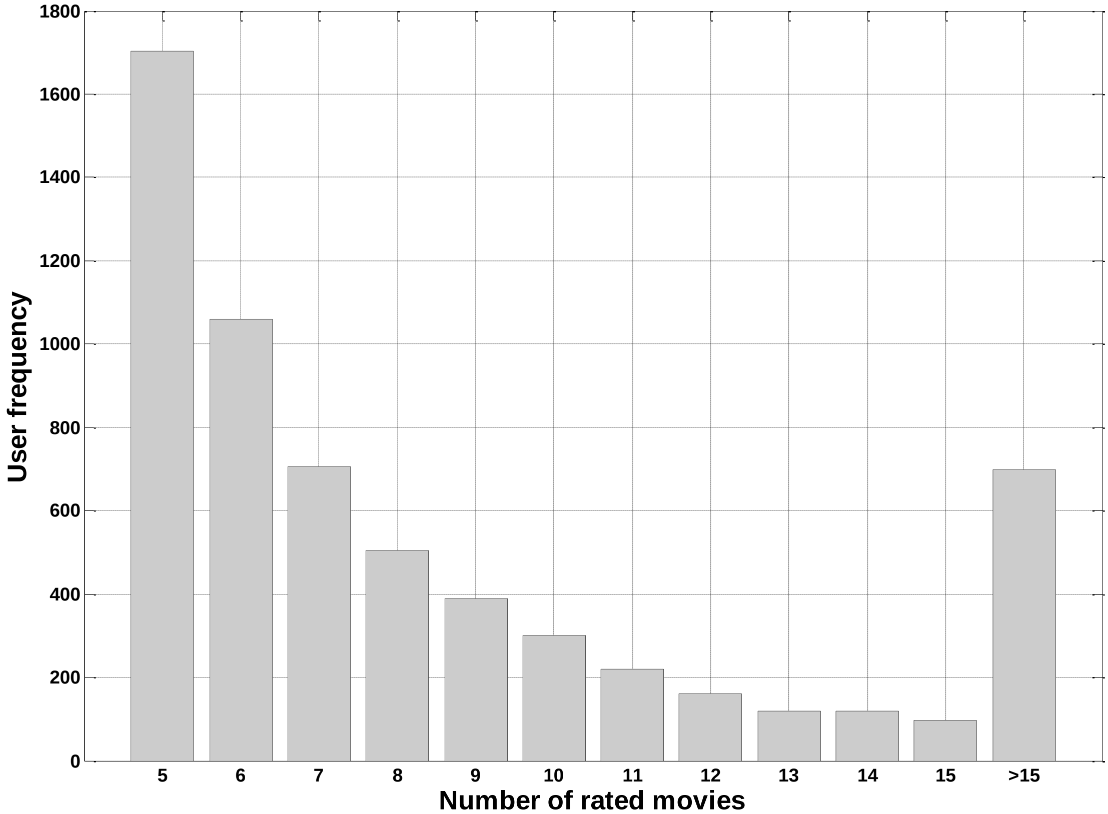


Figure 5.3-5: *Frequency of the number of movies rated by the users of the first data set*

5.3.2 Second data set description and statistics

The data set used in the second evaluation scheme consisted of users that had rated at least 10 different movies, as already mentioned. This experimental data set included 1716 different users that rated 965 movies in total. The overall number of ratings was 34800 and every user had rated about 20 movies on average. The maximum number of movies that a user has rated is 237, the same as in the first data set, since both sets have been derived for the same repository. The average evaluation grade was 9.5, 9.8, 9.4, 10.0 and 9.4 for the criteria *acting, story*, *direction*, *visuals* and *overall*, respectively.

In **Figure 5.3-6** a histogram of the ratings that users provided is shown, for all criteria as well as for the overall preference ratings.

The frequency of rated movies is provided in **Figure 5.3-7**, classified into the 6 different groups as mentioned and described in the first data set. The second class, consisting of those movies that have been rated between 11 and 50 times, seems once more to be the highest populated class, since 432 (~25%) movies belong to this group. The frequency rating ranges between 1 and 722 and the average rating for a movie is about 36.

In **Figure 5.3-8** the distribution of the number of rated movies by every user is provided. By interpreting this distribution, it is noticed that about 17% of users had rated 10 movies while the majority of users, about 40% of the users had rated over 15 movies, a reasonable observation due to the filter applied.

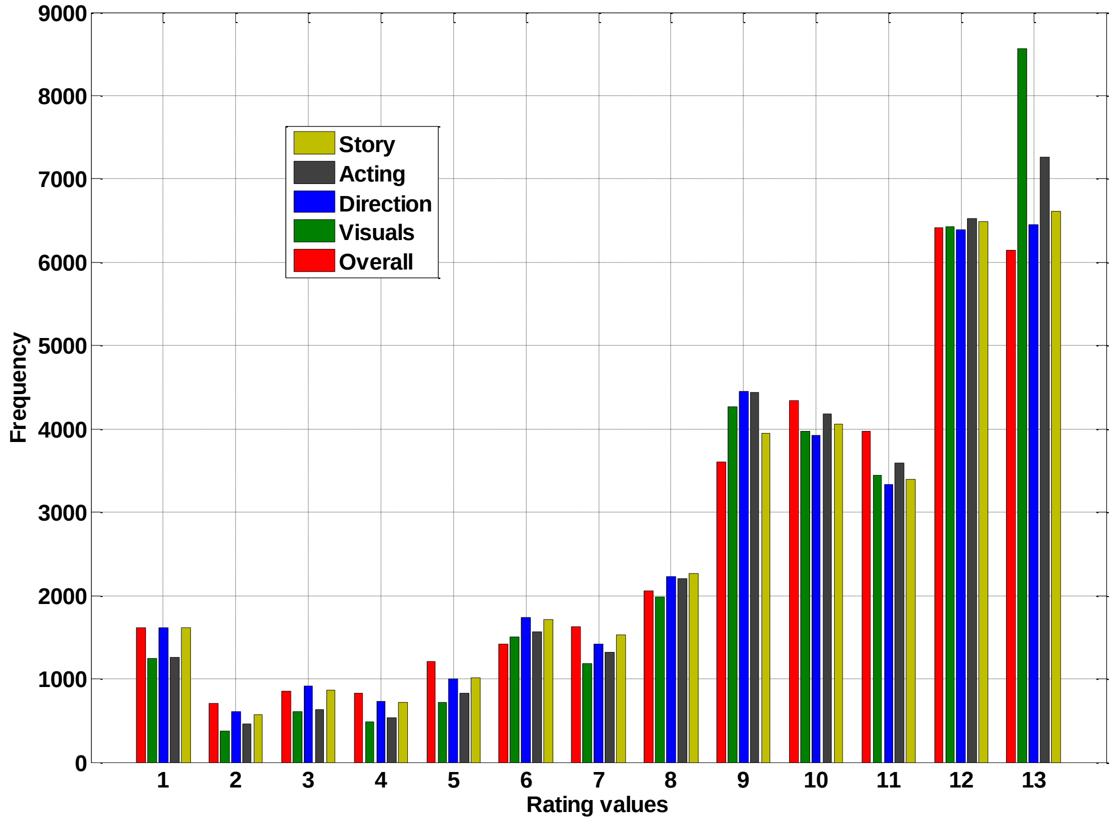


Figure 5.3-6: *Distribution of the data set for users with at least 10 rated movies. The five criteria and the overall statements correspond to the six different color bars as indicated in the legend.*

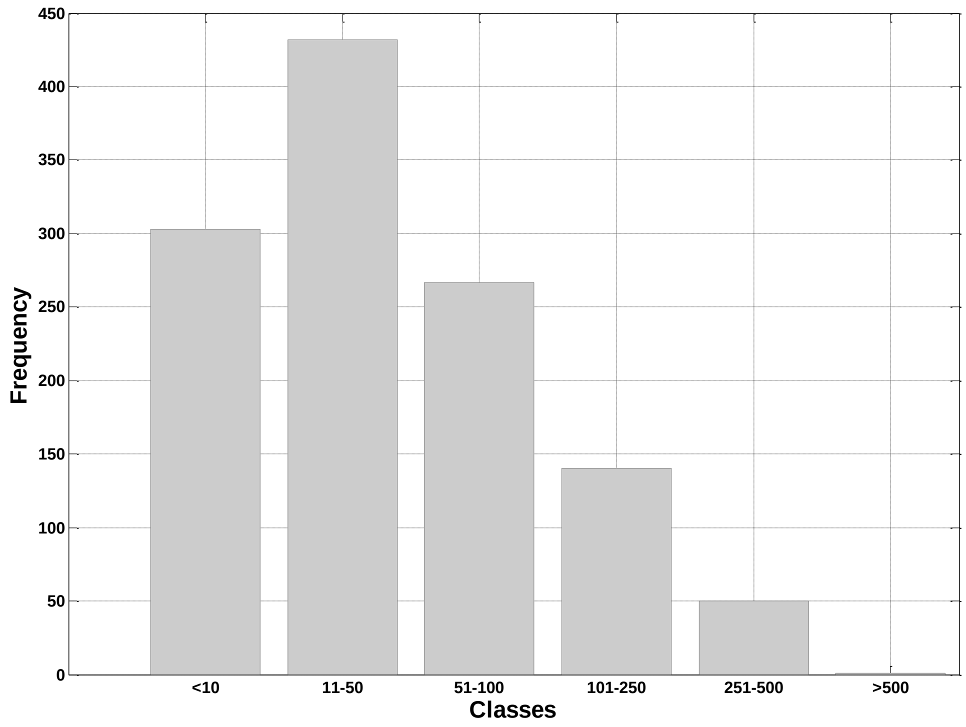


Figure 5.3-7: *Frequency of rated movies per class for the second data set*

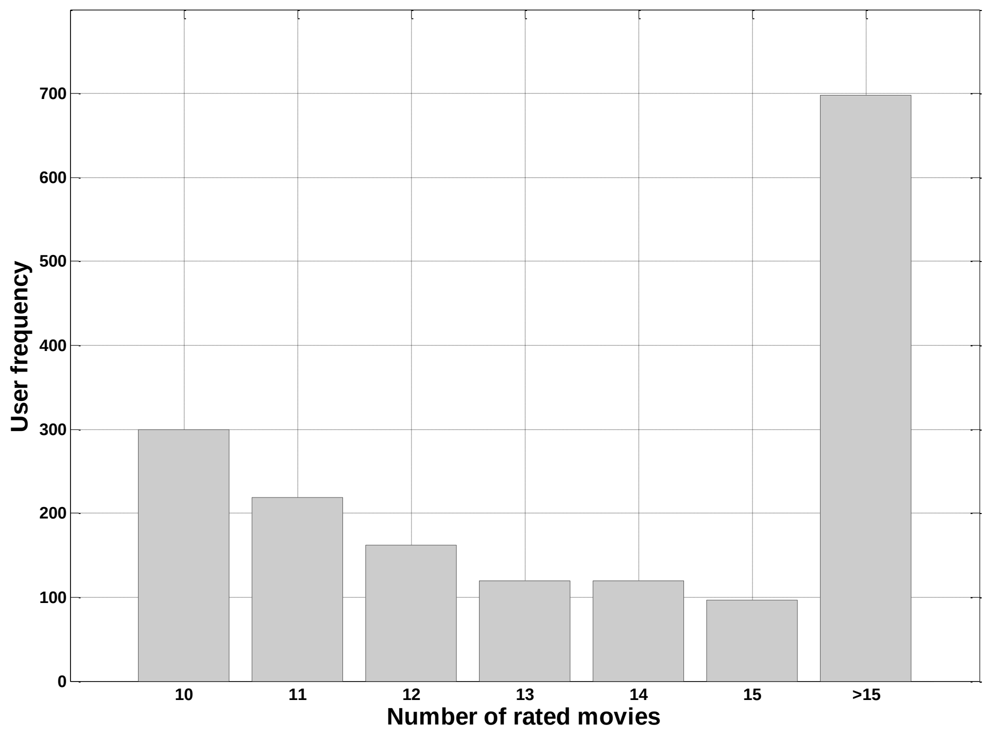


Figure 5.3-8: *Frequency of the number of movies rated by the users in the second data set*

5.3.3 Third data set description and statistics

The data set used in the third evaluation scheme consisted of users that had rated at least 35 different movies, as mentioned above. This experimental data set included 191 different users that rated 917 movies in total. The overall number of ratings was 11757 and every user has rated about 62 movies on average. The maximum number of movies that a user has rated is 237, the same as in the first data set, since both sets are drawn from the same repository. The average evaluation grade was 9.2, 9.5, 9.2, 9.7 and 9.2 for the criteria *acting, story*, *direction*, *visuals* and *overall*, respectively.

In **Figure 5.3-9** a histogram of the ratings the users provided is shown for all criteria as well as for the overall preference.

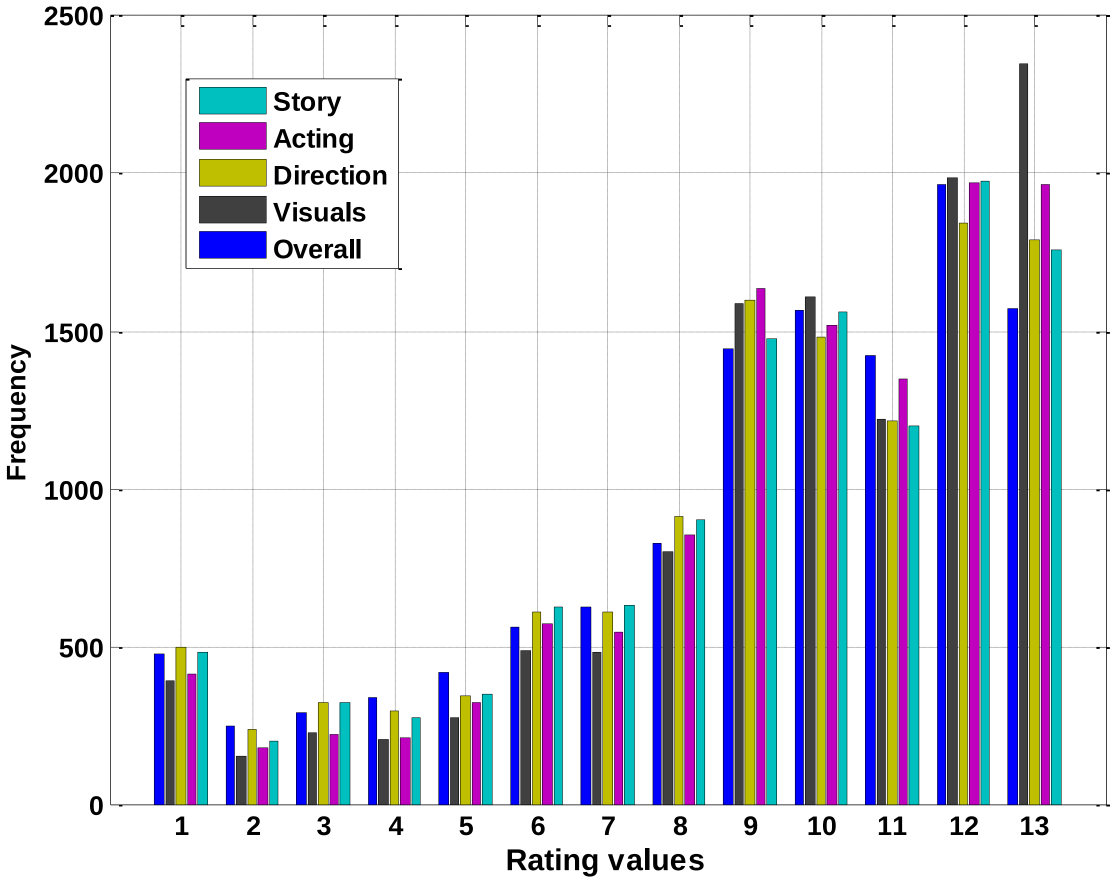


Figure 5.3-9: *Distribution of the data set for users with at least 35 rated movies. The five criteria and the overall statements correspond to the six different color bars as indicated in the legend.*

The frequency of rated movies is provided in **Figure 5.3-10,** classified into the 6 different groups as mentioned and described in the first data set.

It is easily noticed in **Figure 5.3-10** that the first class, which contains movies, rated less than 10 times, appears as the most frequent class in this data set. This is not an unexpected result, since this specific data set includes much less users than the previous two. Therefore, it is not expected the same movie to appear i.e. 100 times in a data set of 191 users. This would mean that many of the users have seen and evaluated several common movies, which is not the case in this data set.

In **Figure 5.3-11** the distribution of the number of rated movies by every user is provided. By interpreting this distribution, it is noticed that about 29% of users had rated 41 to 50 movies while a not negligible percent of users, about 13% had rated over 100 movies.

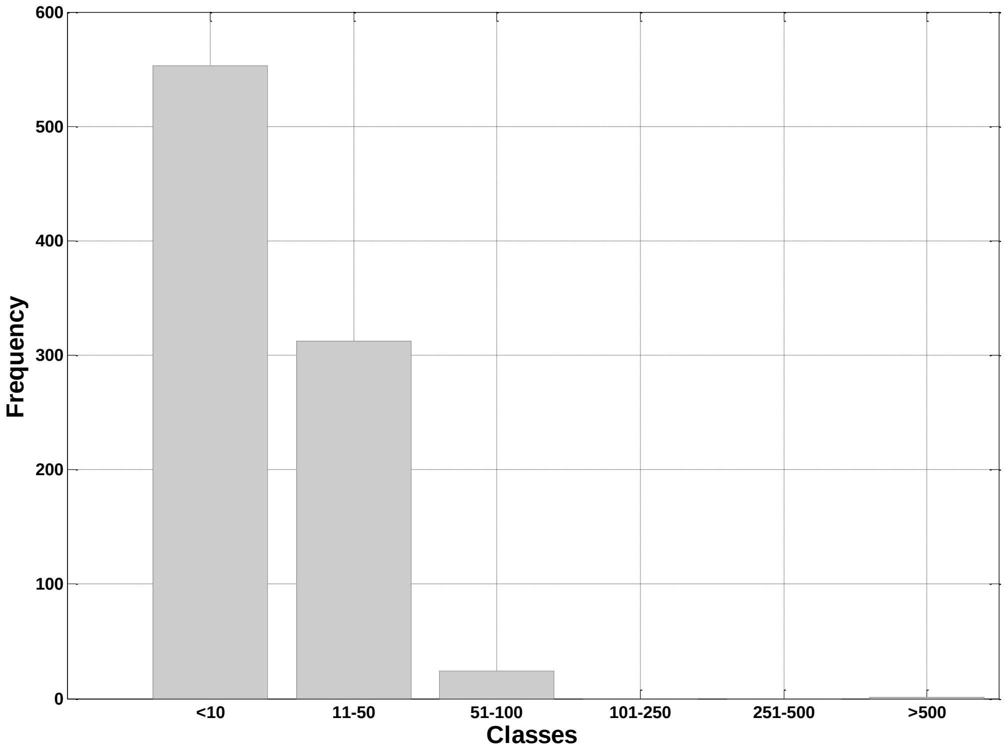


Figure 5.3-10: *Frequency of rated movies per class for the third data set*

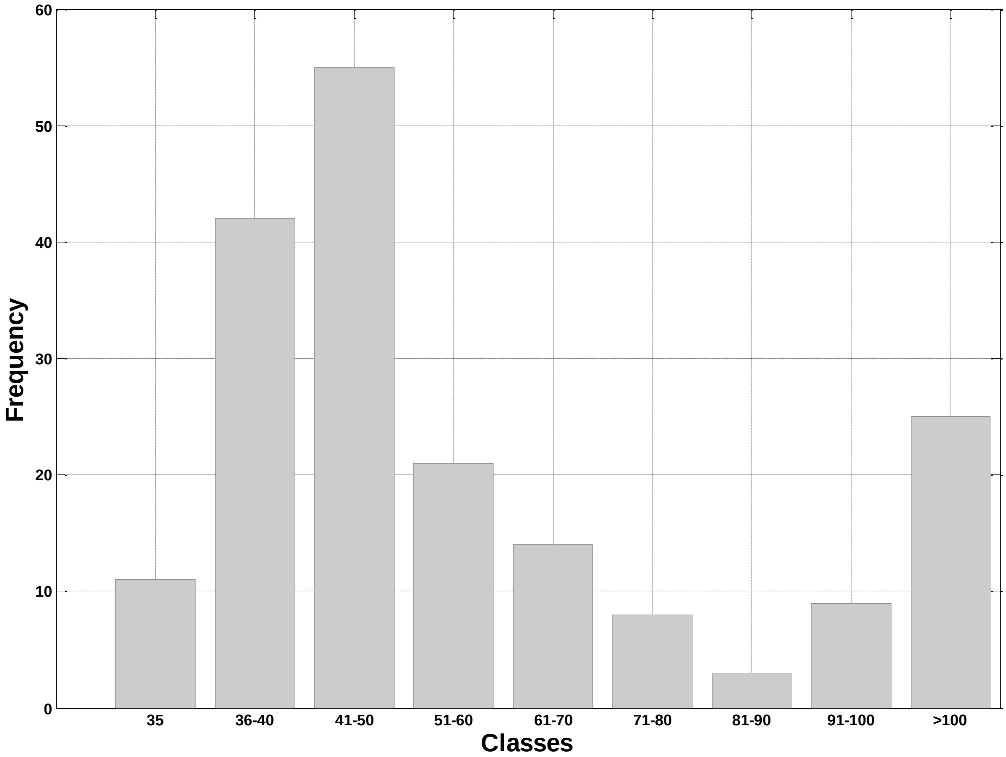


Figure 5.3-11: *Frequency of the number of movies rated by the users of the third data set*

According to the methodological requirements of the Disaggregation-Aggregation approach as discussed earlier in Chapter 3 (3.4.2), a weak preference order of the alternatives is required to apply ordinal regression. The user provided that information, together with the performances on all four criteria for every movie of the reference set. However, because the global preference was expressed in a qualitative scale from 1 to 13, all global preference vales were transformed into a weak preference order for every user. For example, a sequence of numerical values like r_i = [13, 12, 12, 6, 1] when transformed into a ranking order will appear like, r'_i = [1, 2, 2, 3, 4].

Eventually, the multi-criteria data matrix, which acts as an input for the UTA* algorithm, consists of the actual user ratings on all four criteria for the items belonging to the reference set, as well as of a weak preference order for these items. An example of the input multi-criteria matrix is presented in the right side of **Table 5.3-1**.

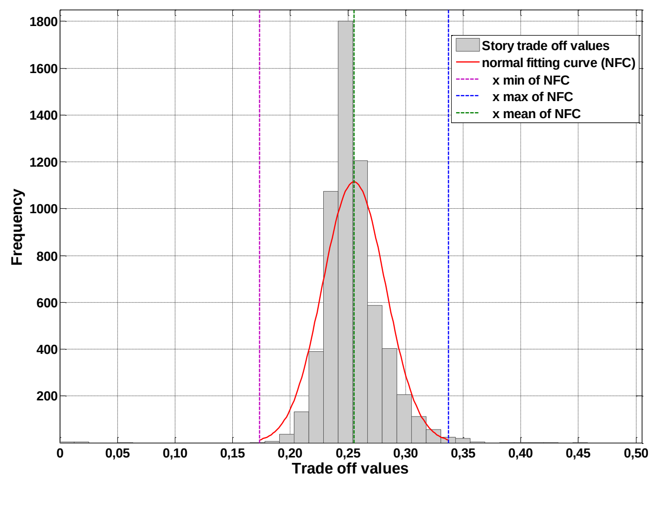
5.4 User modeling phase results

The UTA* algorithm (Y. Siskos, E. Grigoroudis, *et al.* 2005) processed the multi-criteria data matrix to calculate *significance weight vectors* w_u ,for every user u. A matrix of 6078 × 4 was formed in the case of the first data set, a 1716 × 4 in the second data set and a 191 × 4 in the third case, which included the weight vectors of all users. All weights were normalized in a range from 0 to 1. A description of the distributions of all three matrices, to provide a picture together with all necessary statistical parameters follows.

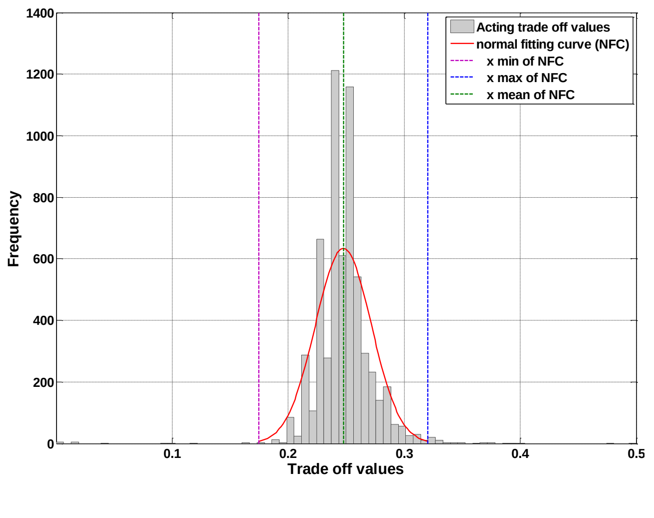
5.4.1 UTA results for the first data set*

As mentioned above, the first data set included 6078 users. Each user had provided preference data for at least 5 different movies, which formed his/ her input (reference set). Independently on the total number of rated movies, UTA* was applied exclusively on the reference set of 5 movies.

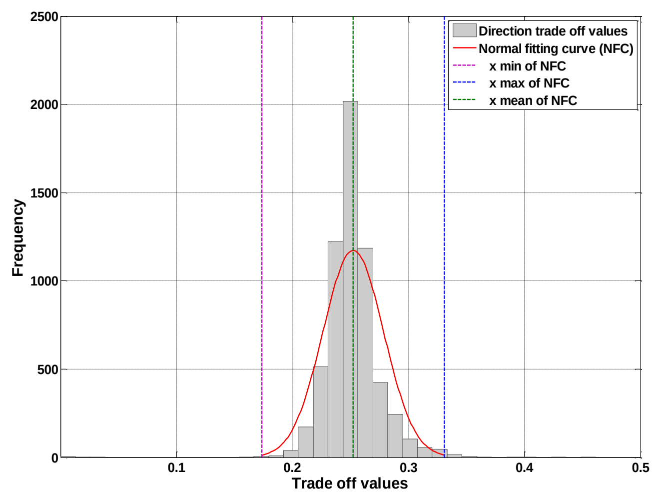
Subsequently, a description of the UTA* results in terms of significance weights is provided in the form of distributions of these significance weights (trade offs).



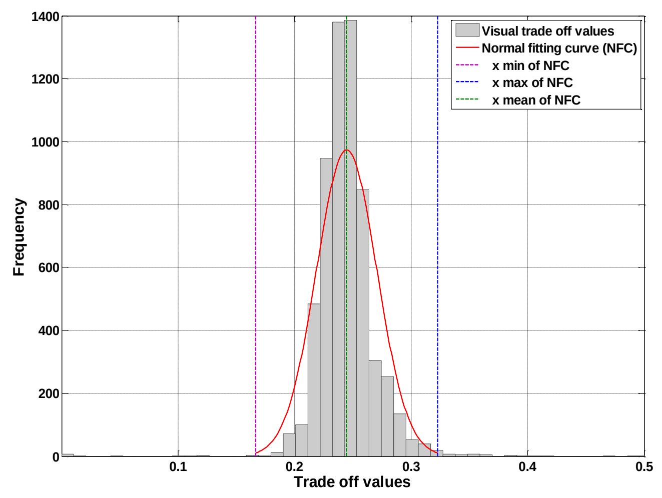
Distribution of trade off values for "Story" in data set 1



Distribution of trade off values for "Acting" in data set 1



Distribution of trade off values for "Direction" in data set 1



Distribution of trade off values for "Visuals" in data set 1

Figure 5.4-1: *Distribution of trade off values for all criteria of the first data set*

	Reference set size=5, 6078 users			
	Story	Acting	Direction	Visuals
Min	0	0	0	0
Max	0.9925	0.5000	1.0000	0.8239
Mean	0.2553	0.2476	0.2524	0.2447
Std	0.0273	0.0242	0.0262	0.0260

Table 5.4-1: *Statistical data for the first data set*

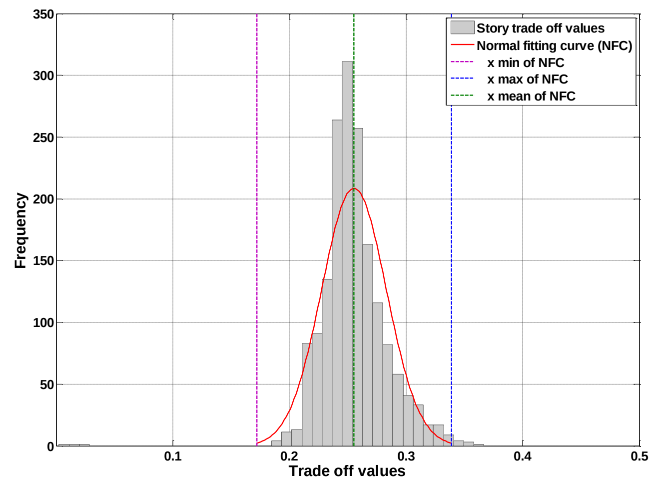
In **Figure 5.4-1** all four distributions of criteria weights (trade offs) are provided fitted by a normal curve. Parameters of the Normal Fitting Curve (NFC) are illustrated by vertical lines on each graph. Left discontinuous vertical line corresponds to the minimum value of the normal fitting curve (magenta), middle discontinuous vertical line (green color) corresponds to the mean of the NFC and finally, the right line shows the maximum of NFC in each case. All graphs of **Figure 5.4-1** correspond to the first data set, of 6078 users that have rated at least 5 movies.

Table 5.4-1 shows statistical data for the same data set. The minimum, maximum, mean and standard deviation values are provided for the ***actual significance weight data*** this time, as opposed to the statistical data illustrated in the graphs that correspond to the NCF. It is noticed from **Table 5.4-1** that the maximum value is given to criterion "direction" while criterion "story" takes on average larger values that the other three, with greater also disperse.

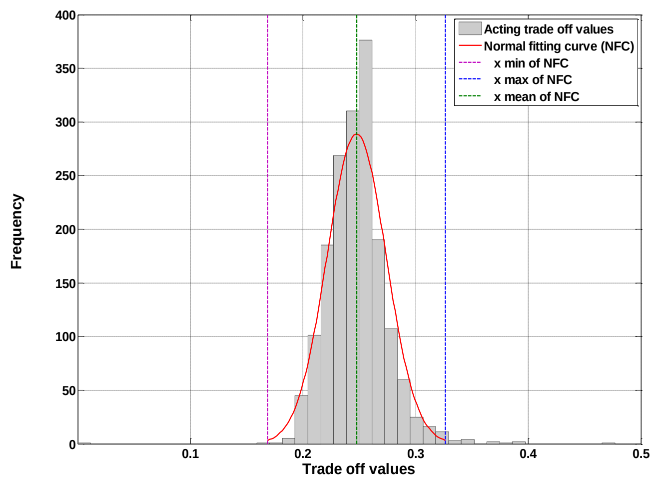
5.4.2 UTA results for the second data set*

As already stated the second data set was used to ensure a more consistent and reliable evaluation analysis complementary to the first one. The important characteristic of this data set is that system's recommendations are evaluated on at least 5 different movies for every user.

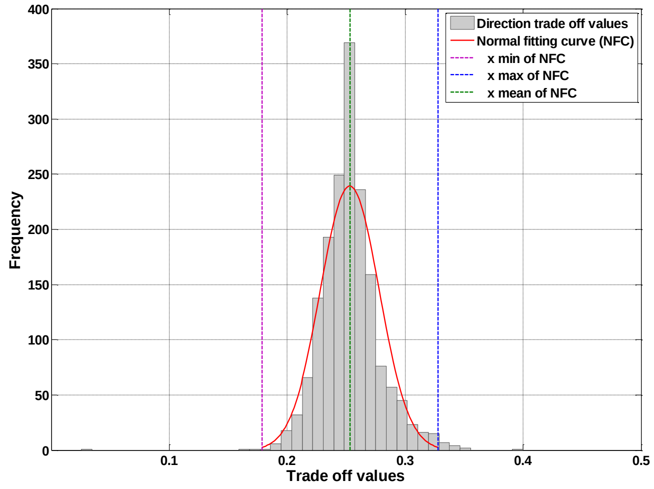
In **Figure 5.4-2** the corresponding to four criteria distribution values are shown and statistical data for the Normal Fitting Curve are illustrated with dashed lines accordingly.



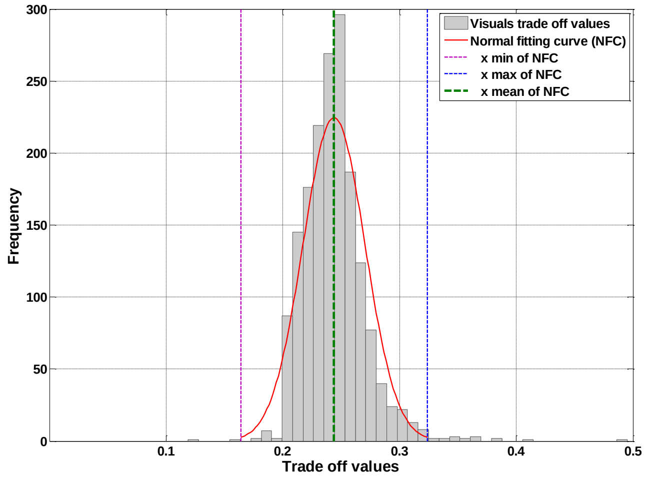
Distribution of trade off values for Story in data set 2



Distribution of trade off values for Acting in data set 2



Distribution of trade off values for Direction in data set 2



Distribution of trade off values for Visuals in data set 2

Figure 5.4-2: *Distribution of trade off values for all criteria of the second data set*

In **Table 5.4-2** statistical data are shown for the actual trade off values. We may once more notice that criterion "Story" has the maximum average value and dispersion.

	Reference set size=5, 1716 users			
	Story	Acting	Direction	Visuals
Min	0.0025	0	0.0255	0.1187
Max	0.3669	0.4769	0.4000	0.4952
Mean	0.2554	0.2476	0.2532	0.2439
Std	0.0278	0.0263	0.0249	0.0267

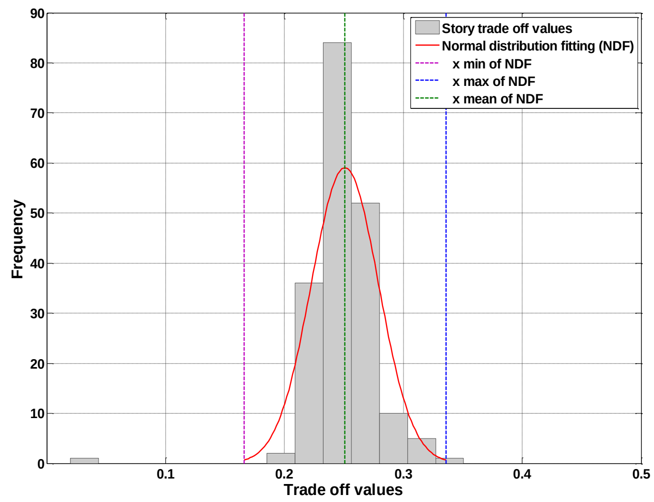
Table 5.4-2: *Statistical data for the second data set*

5.4.3 UTA results for the third data set*

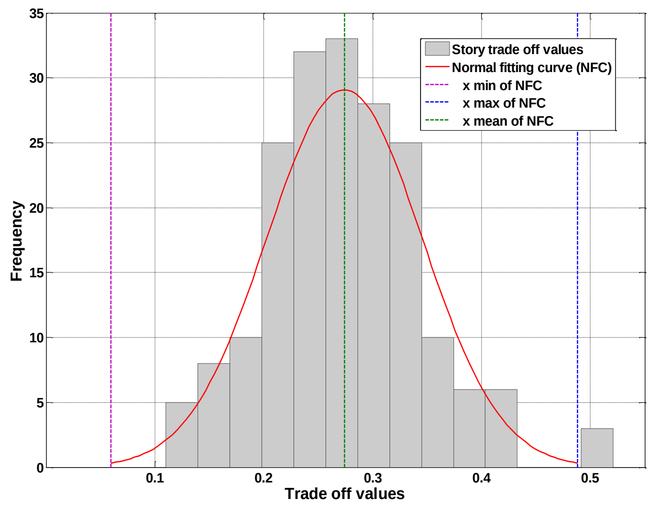
The third data set was designed to study other attributes that may affect system's performance, like the number items that belong to the reference set A_R.

The important characteristic of this data set is that every user that belongs there has rated at least 35 different movies. By filtering out only those users, a bias may be induced, since this third data set consists of the so called “loyal” users, meaning users that have rated many movies and thus provided more preference information to the system. To this end, the reference set size effect is studied solely for those users and all comparisons are made among them.

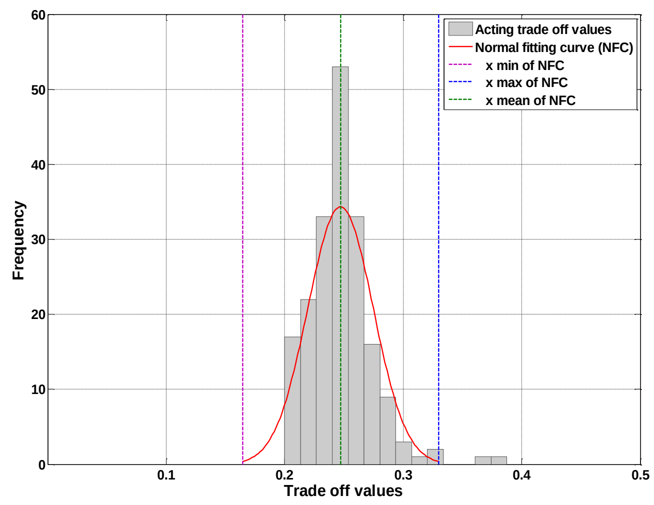
In **Figure 5.4-3** the corresponding to four criteria distribution values are shown and statistical data for the Normal Fitting Curve are illustrated with dashed lines accordingly.



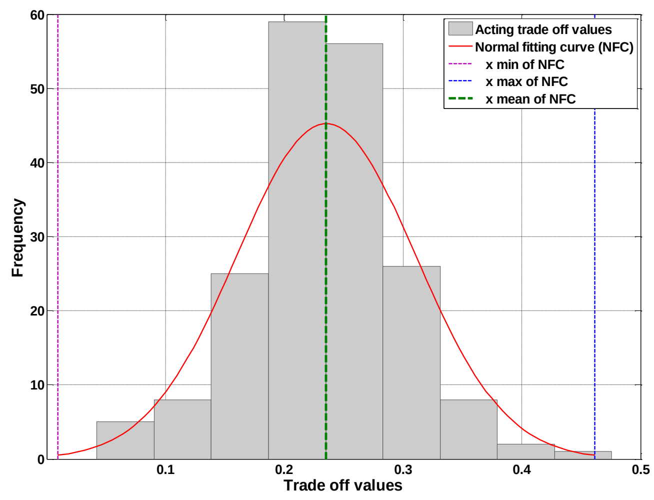
Distribution of trade off values for Story in data set 3, RS=5



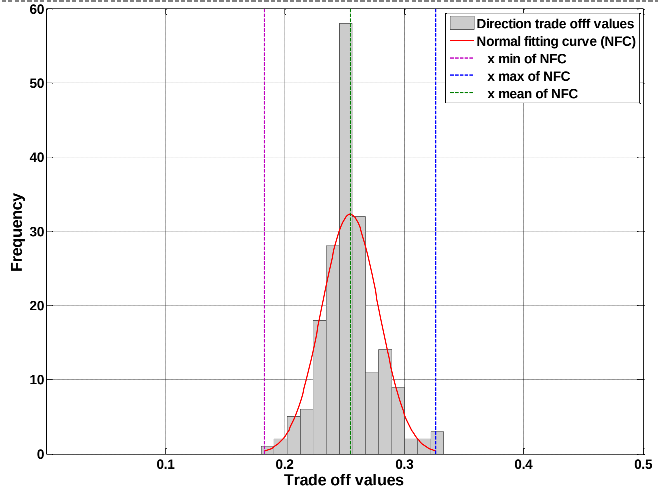
Distribution of trade off values for Story in data set 3, RS=30



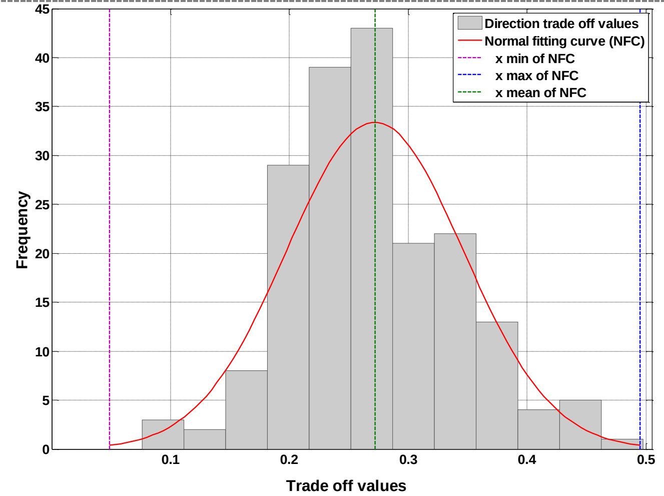
Distribution of trade off values for Acting in data set 3, RS=5



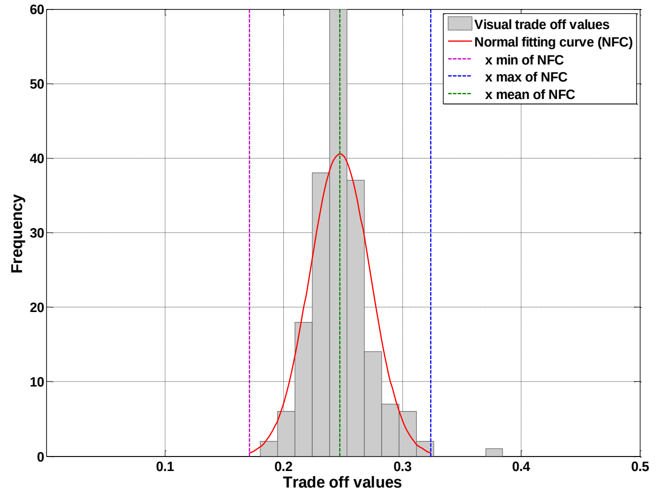
Distribution of trade off values for Acting in data set, RS=30



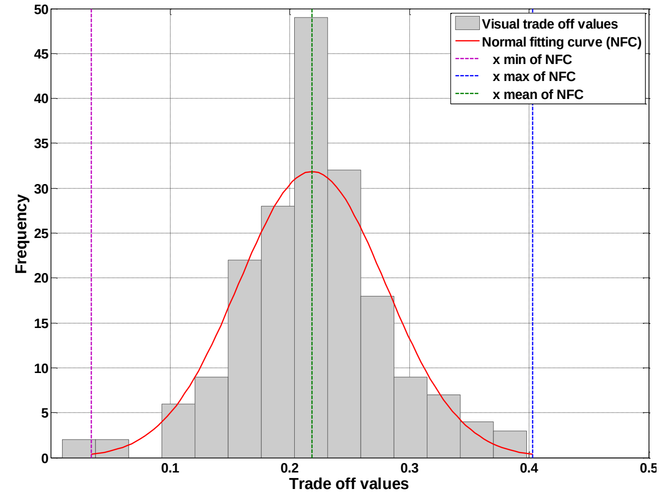
Distribution of trade off values for Direction in data set 3, RS=5



Distribution of trade off values for Direction in data set 3, RS=30



Distribution of trade off values for Visuals in data set 3, RS=5



Distribution of trade off values for Visuals in data set 3, RS=30

Figure 5.4-3: *Distribution of trade off values for all criteria of the third data set*

Since the reference set size effect was studied for four different cases a) when 5, b) when 10, c) when 20 and d) when 30 alternatives are used to model user preferences, it makes no sense to show all the distribution graphs for all cases. Therefore, the distribution curves are shown only for the first and the last case, when the size of the reference set is 5 (RS=5) and 30 (RS=30) accordingly, however statistical data are provided in table for all four different cases, again for the actual data rather that for the Normal Fitting Curve data.

	Reference set size=5, 191 users				Reference set size=10, 191 users			
	Story	Acting	Direction	Visuals	Story	Acting	Direction	Visuals
Min	0.0200	0.2000	0.1800	0.1800	0	0.1468	0.1131	0.1394
Max	0.3508	0.3875	0.3331	0.3846	0.4250	0.3954	0.4079	0.6000
Mean	0.2509	0.2471	0.2547	0.2473	0.2652	0.2372	0.2606	0.2369
Std	0.0283	0.0240	0.0240	0.0255	0.0527	0.0458	0.0466	0.0564
	Reference set size=20, 191 users				**Reference set size=30, 191 users**			
	Story	Acting	Direction	Visuals	Story	Acting	Direction	Visuals
Min	0.1236	0.0547	0.0932	0.0075	0.1100	0.0422	0.0763	0.0100
Max	0.5030	0.4916	0.4825	0.3980	0.5211	0.7167	0.5681	0.3978
Mean	0.2683	0.2331	0.2716	0.2270	0.2741	0.2353	0.2720	0.2186
Std	0.0673	0.0643	0.0633	0.0572	0.0714	0.0753	0.0744	0.0615

Table 5.4-3: *Statistical data for the third data set*

To further interpret the statistical data presented in **Table 5.4-3** and specifically the evolution of the criteria means over the reference set size, the latest is plotted against the reference set size for every criterion and the results are shown in **Figure 5.4-4**. It is shown there that the two criteria “story” and “direction” demonstrate a similar behavior over the increase of the number of alternatives in the reference set, while the remaining two, “acting” and “visuals”, evolve similarly at the short reference set sizes and significantly distinguish themselves only when 30 alternatives are used to model user preferences. The aforementioned behaviors become apparent after the increase of the reference set to include 10 movies.

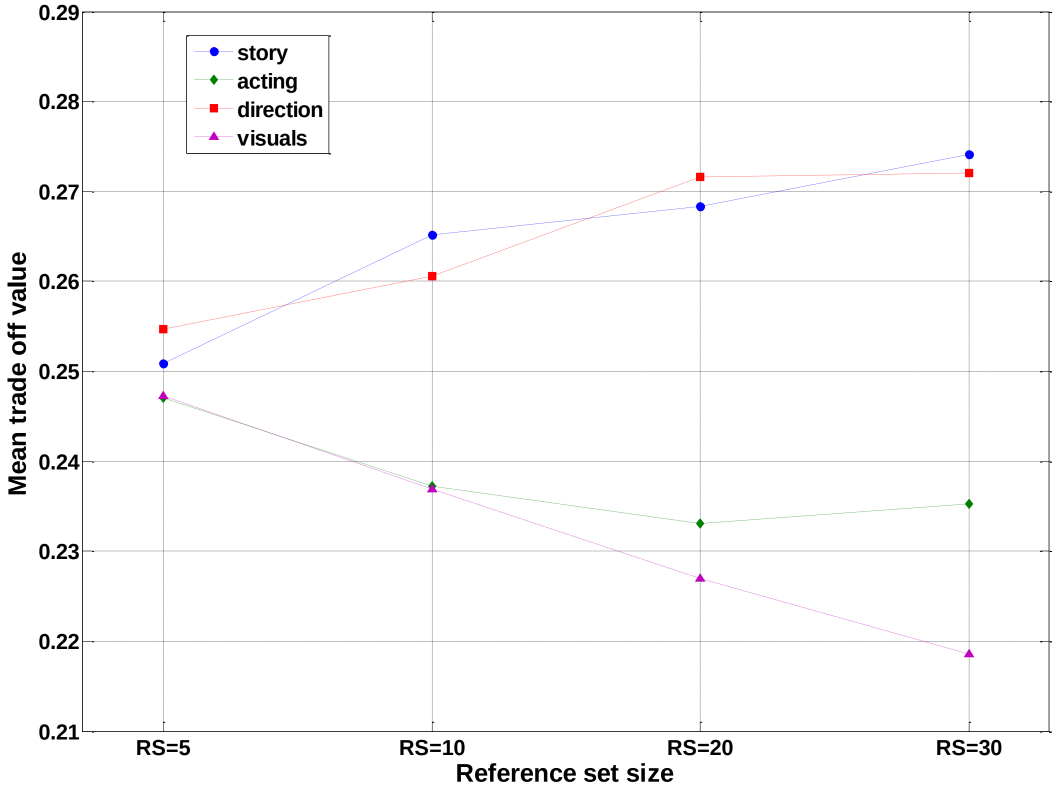


Figure 5.4-4: *Mean trade off values over the reference set size*

5.4.4 Reference set size effect in user modeling

With the aim to reproduce user's initial ranking order, UTA* calculates a set weights (trade offs) for all criteria that when combined with user's ratings on the same criteria it would reproduce user's initial ranking order. To this end, a rational measure to validate UTA* results in terms of how accurately manages to reproduce user's initial ranking order is Kendall's tau. However, another measure that would also indicate a modeling accuracy level of UTA*, is the sum of all errors introduced in algorithm's step 4. The latest, can be considered as a more rigorous, yet reliable measure of UTA*'s performance. In **Figure 5.4-5** the average user sigmas is ploted against different number of alternatives used as a reference set. The reference set size varied from 5 to 35. A total decrease of about 31% from reference set size equals to five, to a reference set size of 30 alternatives, when a plateau seems to be formed, indicates that the number of alternatives plays a crucial role during the user modeling phase.

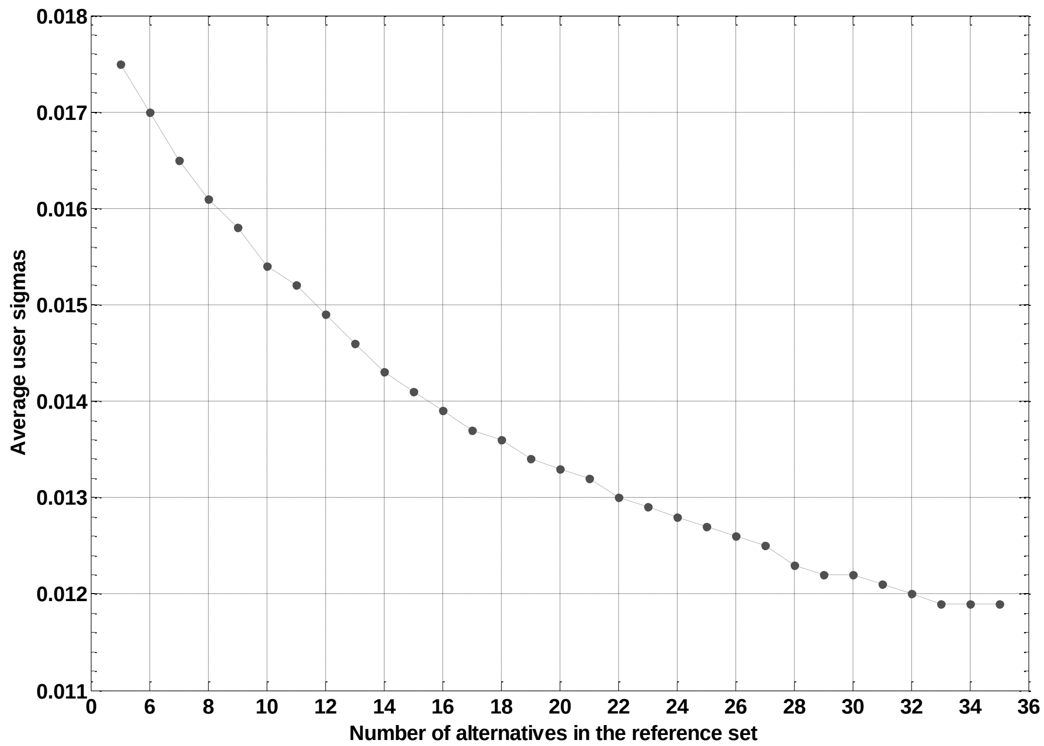


Figure 5.4-5: *Average user sigmas vs. number of alternatives used to model user preferences*

Although a greater reference set leads to more accurate user modeling, it can also be considered as a restrictive factor in the case of Recommender Systems. More alternatives in the reference set means that a user needs to explore and evaluate more items. It is generally considered that users are unwilling to provide a lot of information in a system. Nevertheless, in the proposed system, this can be anticipated during the feedback phase. At first, the system only asks for preference information concerning five different alternatives. However, as the user interacts with the system and provides feedback, his/her profile is updated by including this new alternative in the reference set providing thus, an incrementally more accurate preference modeling.

5.5 Clustering phase results

Global k-means algorithm divided the significance weight vectors, resulted from the user modeling phase, into separate clusters. As already stated, global k-means ensures optimality at each clustering step. This means that the Sum of Squared Error (SSE) will continuously decrease over the number of clusters. Plots of SSE for different number of clusters are shown in **Figure 5.5-1** and **Figure 5.5-2** for the two first data sets respectively.

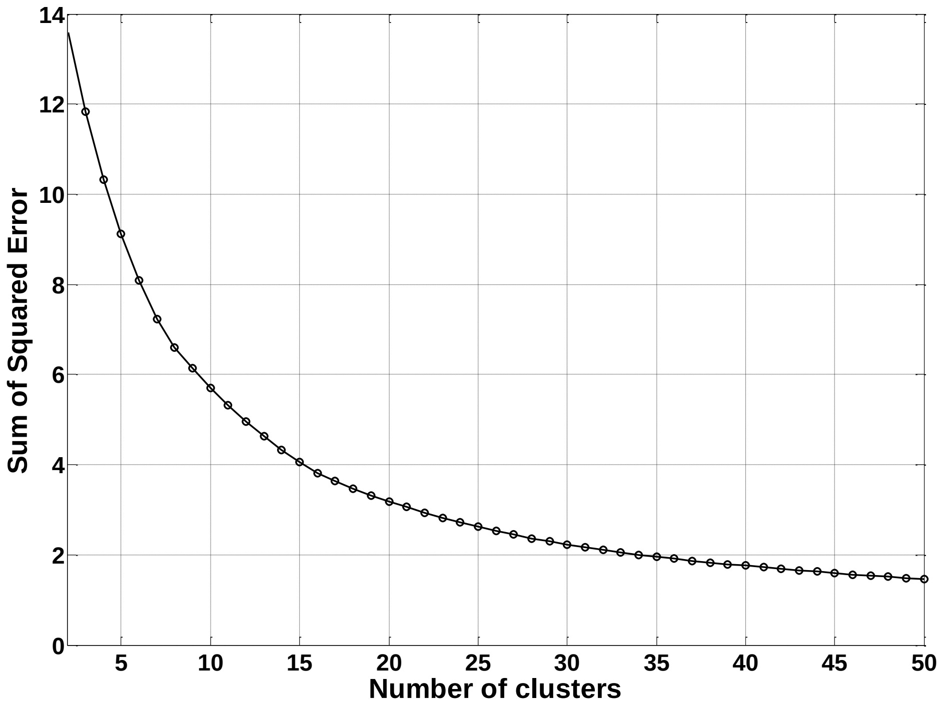


Figure 5.5-1: *Sum of squared errors (SSE) versus the number of clusters for the first data set of 6078 users*

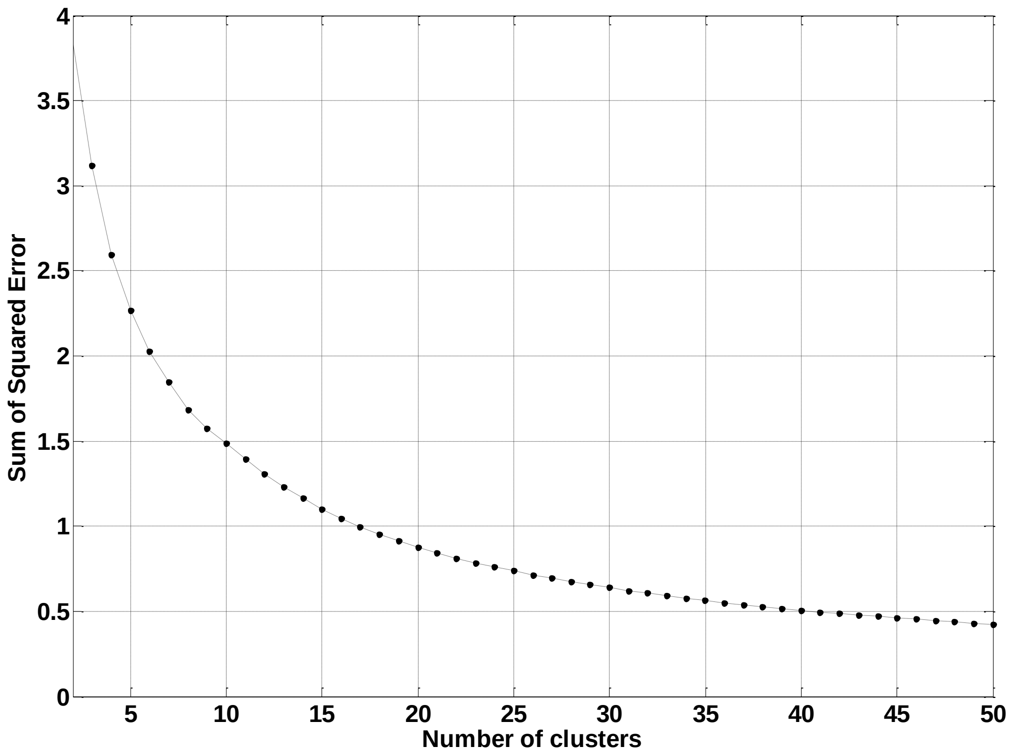


Figure 5.5-2: *Sum of squared errors (SSE) versus the number of clusters for the first data set of 1716 users*

Even though SSE cannot be compared in absolute values for the two different data sets, the relative % decrease from the SSE in specific number of clusters such as 20, 30 and 50 may be calculated. For the first data set, these percentages are 76.6%, 83.6% and 87.9%, accordingly, while for the second data set the same proportions become 77.1%, 83.3% and 88.9% respectively.

In **Figure 5.5-3** the corresponding Sums of Statistical Error during the clustering procedure of the significance weight vectors, versus number of clusters have been plotted for the third data set (191 users). In this case, the number of total users is constant, while the number of movies used to model user preferences varies from 5-30.

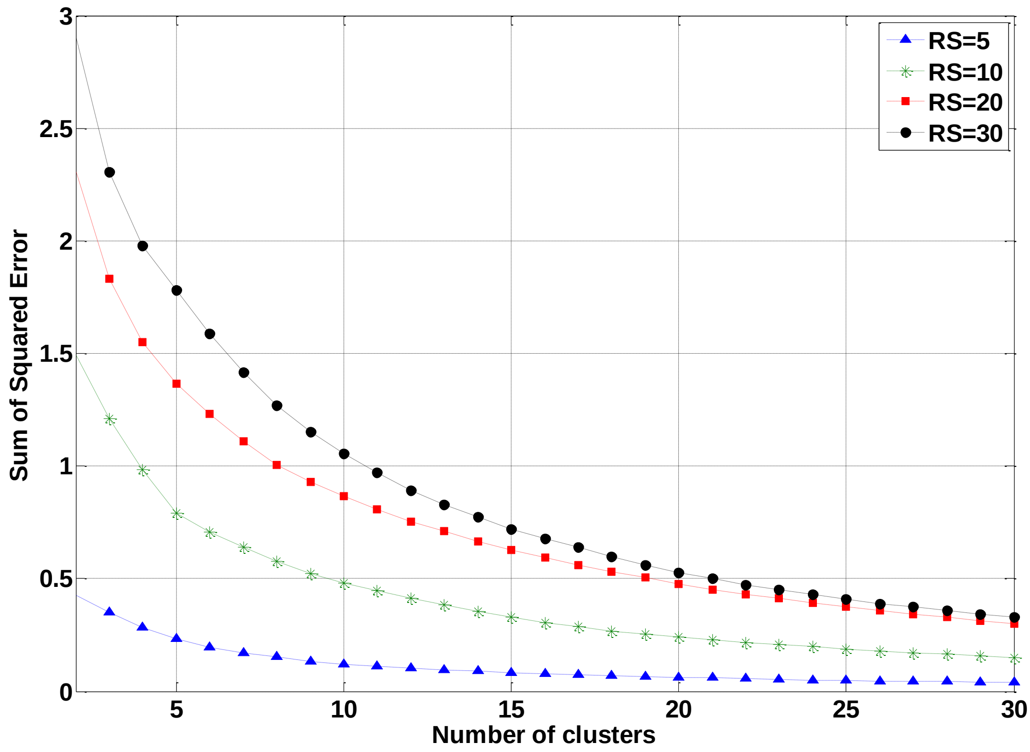


Figure 5.5-3: *Sum of squared errors (SSE) versus the number of clusters for the first data set of 191 users*

	2 to 10 clusters	**2 to 20 clusters**	**2 to 30 clusters**
RS=5	71.7%	85.3%	90.8%
RS=10	67.7%	83.9%	89.8%
RS=20	62.5%	79.3%	87.0%
RS=30	63.7%	81.8%	88.7%

Table 5.5-1: *Percentage improvement of Sum of Squared Error in different stages of the clustering phase for the third data set.*

It is observed from **Figure 5.5-3** that SSE increases as the reference set size increases. This result does not necessarily mean that worse clusters are formed as more alternatives are introduced to the reference set. On the contrary, this effect should be examined together with the fact that in **Figure 5.4-3** it was shown that the dispersion of tradeoffs significantly increases with the reference set size. This can easily explain the increase of SSE in terms of absolute values. Furthermore, **Table 5.5-1** shows a similar result for all cases: the evolution of all curves, independently on the reference set size, indicates that at about 30 clusters the decrease of SSE can be considered significantly adequate in all cases.

Although we can get a rough estimation of the clustering tendency from **Figure 5.5-1, Figure 5.5-2** and **Figure 5.5-3**, since the SSE will be constantly decreasing with the number of clusters, further investigation to identify the "optimal" number of clusters is necessary and most of times is application dependent.

The final outcome of the third phase is a collection of disjoint groups of users with similar preferences. These groups constitute the user profile clusters that the system's final step exploits to provide item recommendations. These groups can be updated when required as explained in the feedback mechanism (see 4.2.5).

Another metric that could act as an evaluation measure in this case, is the user similarity as calculated from equation 4.2.13. It is advocated in this thesis that clustering identifies user groups with similar preferences. This is accomplished by means of multi-criteria user modeling. However, to predict a value for a candidate movie, UTARec employs also the notion of user similarity as inspired by the collaborative filtering philosophy (see 4.2.13 & 4.2.14). As a result, it is expected for the user similarity to grow inside a cluster as this becomes denser. Indeed, the aforementioned hypothesis is verified in **Figure 5.5-4**. The user similarity is calculated according to equation 4.2.13 and these values are subsequently averaged by equation and plotted against the number of clusters. Analytically, equation 5.4.4-1 is used to find the average per cluster user similarity, where *n* is the number of combination of user similarities. Equation 5.4.4-2 is subsequently used to calculate the average cluster similarity and *numb_clust* is the total number of cluster at which the calculation takes place.

$$avsim(i) = \frac{\sum_{u' \in C(u)} sim(u,u')}{n} \qquad \textbf{5.4.4-1}$$

$$avsim\_clust = \frac{\sum_{i=1}^{numb-clust} avsim(i)}{numb-clust} \qquad \textbf{5.4.4-2}$$

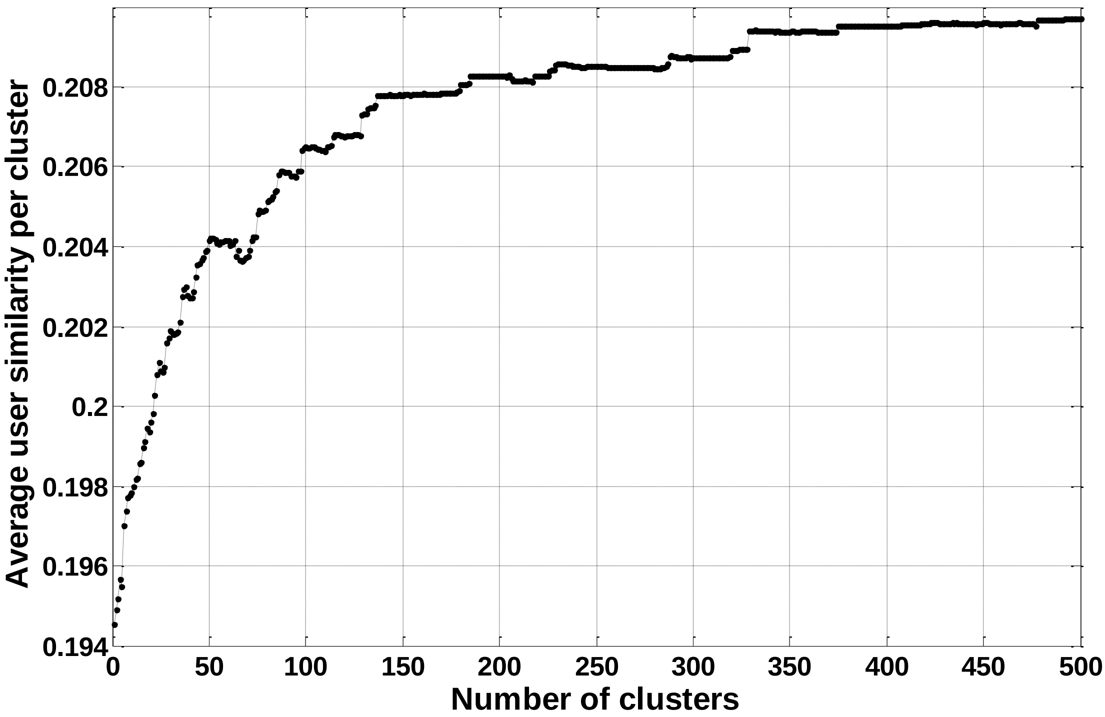


Figure 5.5-4: *Average user similarity per cluster vs. the number of clusters for the data set of 1716 users*

In **Figure 5.5-4** it is shown that after a certain number of clusters this curve reaches a plateau, where no significant change occurs at this region and above. This result indicates that although geometrically accurate clusters are formed even at 30 clusters (see **Figure 5.5-1**, **Figure 5.5-2** and **Figure 5.5-3**), the meaningfulness of clustering varies along applications and thus a plot of SSE versus the number of clusters provides solely an estimation of the efficiency of the clustering process. Further investigation is necessary, to examine the usefulness of the clusters formed. For this reason, a second evaluation analysis follows, that reviews UTARec's performance in several aspects.

A crucial factor that needs to be determined to ensure a complete and straightforward presentation of the evaluation analysis that follows is the number of clusters. It was already mentioned that a plot of the Sum of Squared Error versus the number of clusters provides an indication concerning the quality of clusters formed and an intuition for an "optimal" number of clusters *k*. The correct choice of *k* however is often ambiguous, with interpretations depending on the shape and scale of the distribution of points in a data set and the desired clustering resolution of the user.

Figure 5.5-4 provides a further estimation of the clustering quality by assigning it to the user similarity increase. The total similarity increase that is achieved at about 500 clusters is about 7.8%. More than half of it, 4% and 5% increase was already gained as soon as 30 or 50 clusters were formed, respectively. This means that there is no need in consuming computational recourses to form clusters, when no significant gain is achieved.

To further examine this issue, the Mean Absolute Error was plotted against the number of clusters, as a more appropriate accuracy measure, focused on the prediction efficiency of the recommendation algorithm.

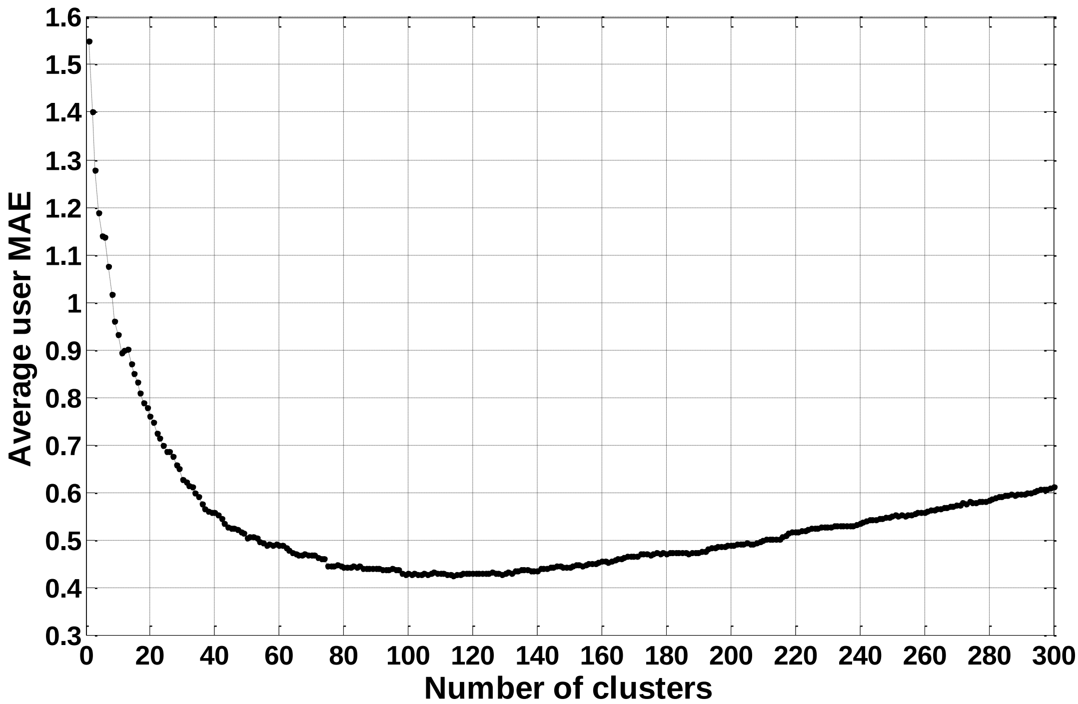


Figure 5.5-5: *Average user MAE vs. number of clusters for the second data set of 1716 users*

It is obvious from **Figure 5.5-5** that the recommendation accuracy of UTARec increases as the number of clusters increases until about 115 clusters are formed. Thereafter, MAE is increased, denoting the degradation of the recommendation accuracy. The maximum increase of the recommendation accuracy, according to MAE results is about 72% relative to the initial MAE that corresponds to ungrouped data. Again, more than half of the total improvement 60% and 67% has been already achieved at 30 or 50 clusters respectively. For this reason, the following evaluation analysis is performed for a total of 30 clusters in most cases, with some refers to the events of 50 clusters.

5.6 Recommendation phase results

During the user modeling phase a reference set of five movies (in the case of the first two data sets) constituted the *training set* A_R of each user. The residual rated movies were used as a *test set* A_T. Thus, during the recommendation phase a rating $R(u,i)$ for every "unseen movie" i, $i \in A_T$ of user u was calculated according to recommendation phase equations (see 4.2.4 for more details).

The presented methodological framework, as analytically described throughout Chapter 4, is compared to several traditional collaborative filtering techniques to demonstrate its accuracy and efficiency. The two statistical accuracy metrics, MAE and RMSE, as well as precision, recall and Kendall's tau were calculated, in every case.

In this part of evaluation analysis, focus is given on recommendation accuracy, which means that differences between recommended and actual values were measured as described hereupon, as opposed to the second type of evaluation analysis provided, the overall functional evaluation analysis, which also considers additional aspects of the system for example the personalization flexibility.

All performance measures were applied on the test set, as previously discussed. The test set for this specific analysis included users that had rated at least one more movie that their related reference set, meaning users that had rated at least 6 movies. The test set included 4376 users in total and each user had rated from 1 to 232 movies, additionally to its reference set. The average number of rated movies was about 7 and the distribution of the number of rated movies for the test set is shown in **Figure 5.6-1**.

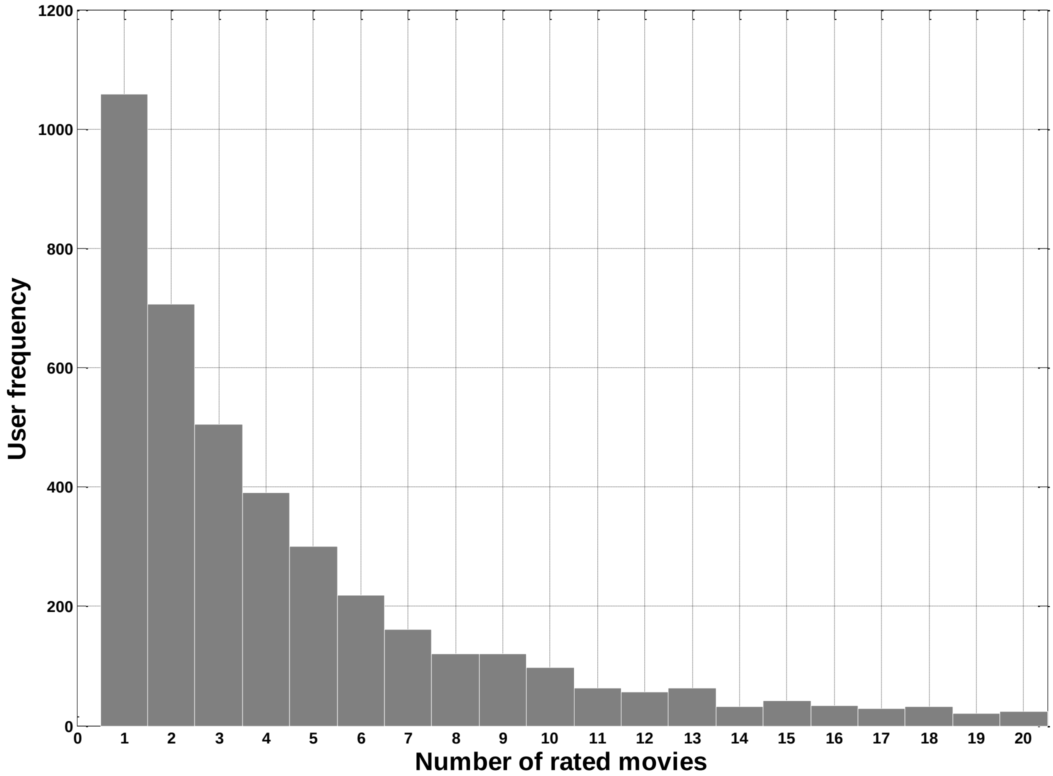


Figure 5.6-1: *Frequency of the number of movies rated by the users*

First, system's performance is calculated based on statistical measures such as the Root Mean Squared Error (RMSE), a very common measure in Recommender System evaluation analysis, which finds differences between system's predicted values and actual values (see 2.6.1).

In **Figure 5.6-2** the RMSE is plotted against the number of clusters formed respectively. In order to calculate the RMSE at a specific number of clusters, first all user RMSEs are averaged for every cluster and the resulted values are subsequently averaged for the cluster.

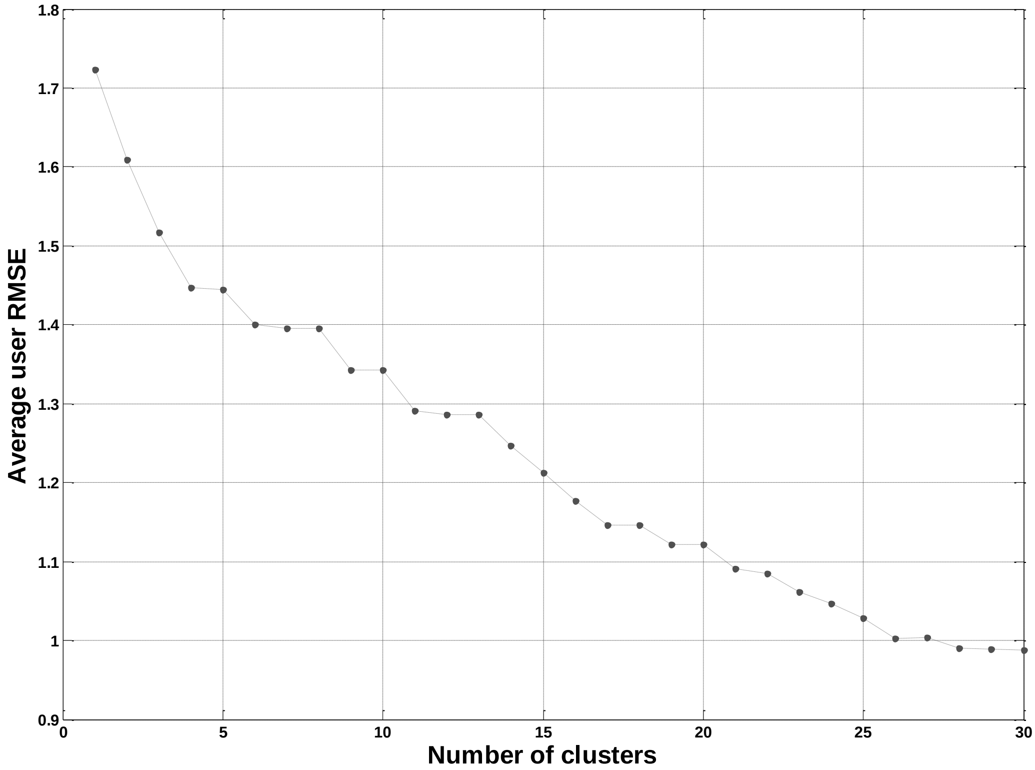


Figure 5.6-2: *Average user RMSE versus number of clusters*

Figure 5.6-2 shows that a relative 42% decrease in RMSE is achieved when 30 clusters are formed.

In **Figure 5.6-3**, the average per user RMSE for an indicative part of users (700-750) is plotted to provide an idea of how RMSE values distribute over different users. Bars correspond to the RMSE values as a result of the application of proposed movie recommender system (when 30 clusters are formed), while the bullets correspond to RMSE values for ungrouped data which in turn correspond to a traditional Multidimensional Collaborative Filtering System.

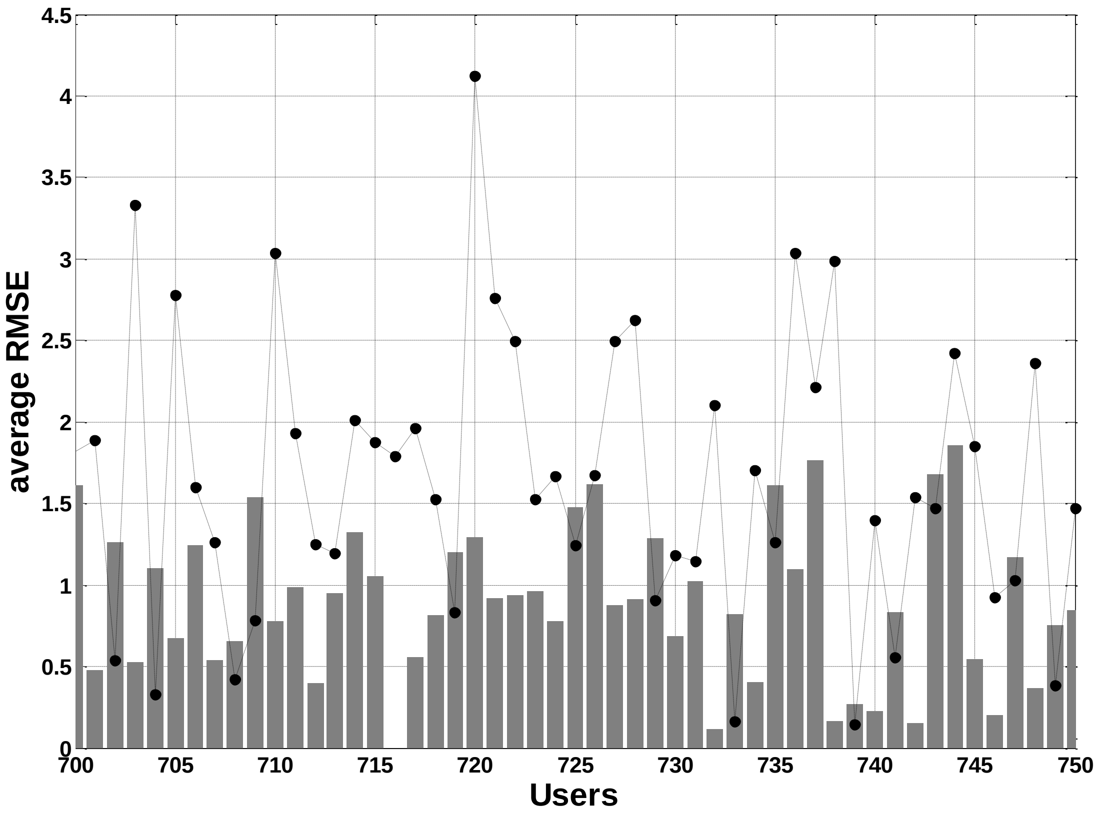


Figure 5.6-3: *Average RMSE for 50 randomly selected users. Grey bars correspond to the average per user RMSE values when 30 user profile clusters are formed and bullets denote the equivalent values for ungrouped data.*

Empty values in **Figure 5.6-3** correspond to users that had rated only 5 movies and thus no recommendation evaluation value can be calculated for them, since their test set is empty.

It is obvious in **Figure 5.6-3** that even if, on average UTARec predicts values close to what users really provided, there are some users where the system falls to predict the correct value. Further and careful examination, which also comprises a primal direction for future research of this thesis, is necessary to study the stochasticity and cause of this phenomenon. A per user analysis would reveal more parameters that should be considered to optimize systems' performance. One possible cause of this deficiency is the lack of a complete family of criteria used to model this specific user.

To effectively apply precision and recall measures, the rating scale was divided into two classes. The first, the *"highly recommended"* class, included only ratings ranged from 11 to 13, while all residual ratings belonged to the second class, the so called *"not recommended"* class. Notice here, that 42% of the overall preference values in our test set fall into the *"highly recommended"* class and so a precision of 42% corresponds to the threshold of a random guess.

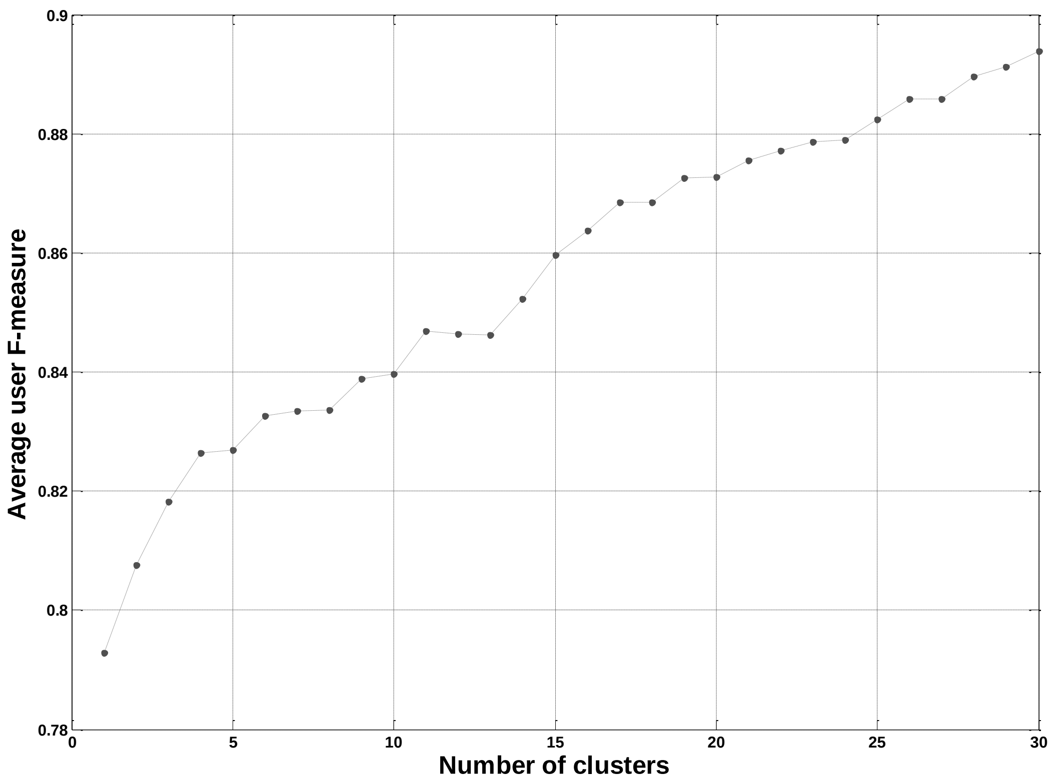


Figure 5.6-4: *Average user F-measure versus number of clusters*

The results of an analysis based on the classification F-measure are shown in **Figure 5.6-4**. There, is shown that a relative improvement of 13% in F-measure is achieved at 30 clusters.

To apply Kendall's tau, the values that UTARec predicted for all "unexplored" items *(test set)* of *u,* were transformed into a user's ranking order, r_u. The same logic was applied to the real ratings of this user and the two ranking orders were compared by Kendall's tau formula (see 2.6.3). The results are shown in **Figure 5.6-5.**

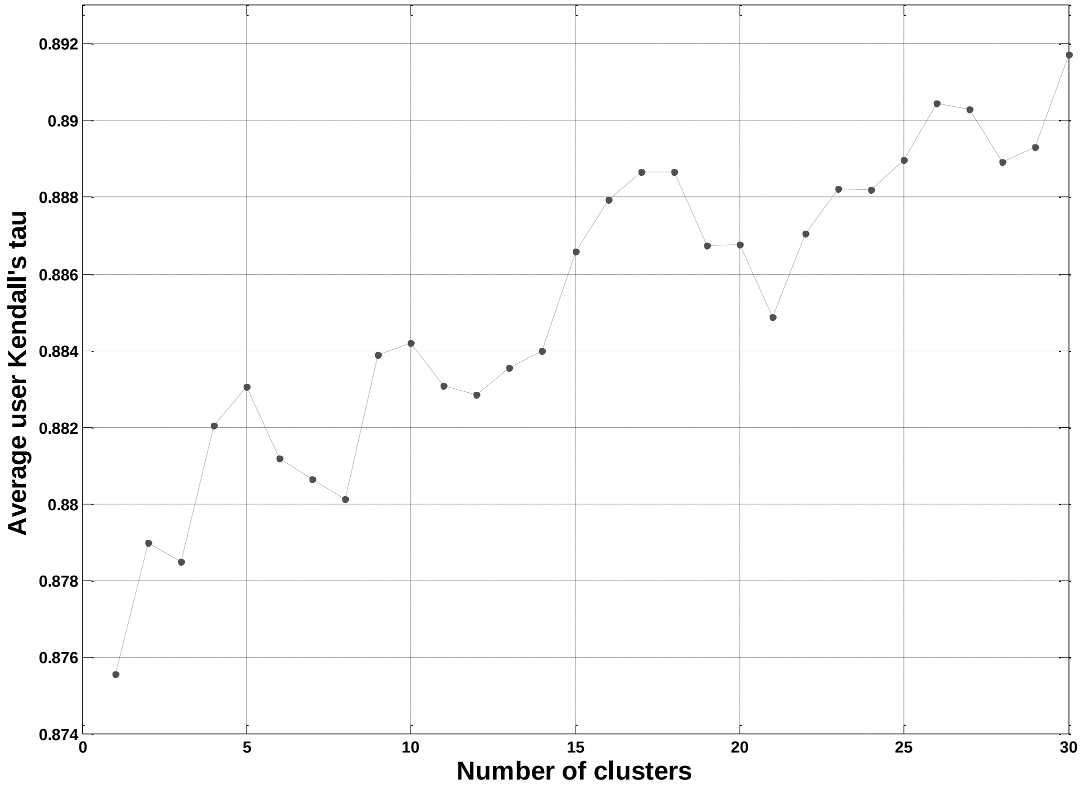


Figure 5.6-5: *Average user Kendall's tau versus number of clusters*

A relatively partially different behavior appears in the case where Kendall's tau is used as an evaluation measure. Overall, this measure indicates a relative 2% increase, the smallest improvement among all evaluation measures.

5.7 Comparison with other recommendation methods

The contribution of UTARec, as a system that demonstrates the multiple criteria framework described in Chapter 4, can be summarized in the following:

- Multiple criteria should be considered to better understand user decision policy, with the purpose of constructing a decision model and be able to recommend items of interest by exploiting this model.
- User modeling phase can be more effective if more sophisticated methods, specially developed to treat multiple criteria decisions, are used to build user profiles. It is important to understand how the user came up to a decision and not only consider his/ her past actions or other people's

similar decisions. It is very likely that two users that gave the same overall grade to an item, to passed through much unlike decision routes to reach at the same point.

- Different users also have different knowledge, interests, abilities, learning styles, and preferences; this however, does not preclude the existence of discrete patterns among users, the so called user profile groups.

As a consequence, in order to verify the effectiveness of UTARec as a multiple criteria Recommender System, its results are compared to popular collaborative filtering approaches as described hereupon. More analytically, the proposed system is compared to: **a)** a *single rating collaborative filtering approach (SR-CF)* that uses the *item weighted cosine-based similarity* to calculate similarities, **b)** a *multiple rating collaborative filtering* technique that uses *average similarity* to aggregate *item weighted cosine based similarities* from individual criteria *(MRCF-av),* and **c)** a MRCF technique that uses *worst-case similarity* to aggregate *item weighted cosine based similarities* from individual criteria *(MRCF-min).*

5.7.1 Single rating collaborative filtering approach (SR-CF)

Single-rating means that only the overall rating is considered in all calculations. In *SR-CF* approach, a similarity index *sim(u,u')* is calculated for all possible *u-u'* combinations according to the *item weighted cosine similarity function* that uses the common notion of cosine similarity. However, even though cosine similarity measure has been extensively used in Recommender Systems, it fails to compute a rating in the case of a single common item. If the number of common items is 1, cosine similarity will result in 1, regardless of the differences in individual ratings. Furthermore since cosine similarity does not consider the size of *U(u,u'),* we used a weighted approach of this measure as given by 5.7.1-1.

$$sim(u,u') = A \cdot \left(\frac{\sum_{i \in U(u,u')} R(u,i) \cdot R(u',i)}{\sqrt{\sum_{i \in U(u,u')} R(u,i)^2} \cdot \sqrt{\sum_{i \in U(u,u')} R(u',i)^2}} \right) \quad \textbf{5.7.1-1}$$

$$A = \frac{U(u,u')}{U(u)} \quad \textbf{5.7.1-2}$$

In equation 5.7.1-1, the traditional cosine similarity measure inside the parenthesis, is multiplied by *A,* which is the percentage of common items *U(u,u')* that both *u* and *u'* have rated, over *U(u),* the total number of items that *u* has rated. Obviously, similarities of pairs of users with large common item set are favored.

5.7.2 Multi-rating collaborative filtering approaches (MRCF)

Two basic approaches have been proposed that include multi-rating information in similarity calculations (G. Adomavicius and Y. O. Kwon 2007). The first considers individual similarities on different attributes, which are traditionally calculated by cosine similarity metrics and the second calculates similarities based on multidimensional distance metrics. The latter, is also adapted in the recommendation phase of the proposed methodology.

Following the first approach, various techniques are employed to aggregate individual similarities. Two different aggregation are applied herein, the *average similarity* and the *worst-case similarity.* Both calculate cosine similarities on all criteria as well as on the overall values. Their only difference is that in the *average similarity* approach these individual similarities are averaged, while in the *worst-case similarity* scenario the minimum of all attribute and overall similarities is chosen to represent users' similarity. However, even though these two approaches are employed as also mentioned in (G. Adomavicius and Y. Kwon 2007), cosine similarities in this work use the *item weighted* variation of cosine similarity as given in equation 5.7.1-1.

Specifically, let's assume that each rating user *u* gives to item *i* consists of an overall rating r_0 and *k* multi-criteria ratings $r_1, \ldots, r_k$:

$$R(u, i) = (r_0, r_1, \ldots, r_k)$$

Then, $k+1$ different similarity estimations can be obtained by using some standard metric to measure similarity between users, *u* and *u'*. In this case $k=4$, therefore 5 different similarities are formed; $sim_0(u, u')$ represents the similarity between u and u' based on the overall rating; $sim_1(u, u')$ represents the similarity based on the first criterion rating; $sim_2(u, u')$, similarity based on the second criterion rating; and so on. The overall similarity is then computed by aggregating the individual similarities. In the case of *average similarity* equation 5.7.1-1 is applied, while in the *worst-case similarity* scenario equation 5.7.1-2 is applied.

$$sim_{avg}(u,u') = \frac{1}{k+1}\sum_{i=0}^{k} sim_i(u,u') \qquad \textbf{5.7.2-1}$$

$$sim_{\min}(u,u') = \min_{i=0,\dots,k}[sim_i(u,u')] \qquad \textbf{5.7.2-2}$$

Initially, the entire experimental data set is encountered as one group and all different collaborative filtering approaches are applied to it. **Table 5.7-1** shows the results of these approaches, first applied to ungrouped data and then to two different stages of the user modeling procedure, after 30 and 50 user profile clusters were formed. *SR-CF* corresponds to a single rating approach discussed earlier, and *MRCF-min* and *MRCF-ave* to the two multi rating collaborative filtering techniques discussed right above. *MRCF-dim* corresponds to the multidimensional collaborative filtering technique that is already included and described in the proposed methodology. All predicted values emerge by applying equation 4.2.4-4 in different notions of similarity, depending on the method used. All similarity *sim(u,u')* and potential rating *R(u,i)* calculations, were implement in the *training set* of 6078 users, while evaluation metrics were calculated over the *test set* of the same users.

In **Table 5.7-1** with **bold** the best performances of all approaches for every metric are signed, while worst values are denoted in *italic*. Moreover, the part of **Table 5.7-1** that corresponds to ungrouped data, serves as a baseline to compare the performance of all methods on different clustering schemes (K. Lakiotaki, N. Matsatsinis*, et al.*).

It is easily observed from **Table 5.7-1**, that among all methods, *MR-CF-dim,* which is the collaborative filtering method that is adapted in the proposed system, outperforms all other methods. It is also evident from **Table 5.7-1**, that user profile clusters improve the performance of any collaborative filtering method.

	MAE	RMSE	Precision	Recall	F-measure	Kendall's τ	
SR-CF	*2.464*	*2.733*	0.8341	*0.283*	*0,423*	*0.865*	*Ungrouped data*
MR-CF-min	2.358	2.620	*0.829*	0.337	0,479	0.869	
MR-CF-ave	2.430	2.695	0.830	0.299	0,440	0.867	
MR-CF-dim	**1.490**	**1.724**	**0.9163**	**0.7416**	**0,820**	**0.876**	
SR-CF	*1.772*	*2.103*	0.882	*0.537*	*0,667*	*0.870*	*30 user profile clusters*
MR-CF-min	1.696	1.992	*0.880*	0.557	0,682	0.875	
MR-CF-ave	1.761	2.077	*0.880*	0.540	0,669	0.873	
MR-CF-dim	**0.819**	**0.956**	**0.964**	**0.843**	**0,899**	**0.889**	
SR-CF	*1.451*	*1.798*	0.905	*0.630*	*0,743*	*0.878*	*50 user profile clusters*
MR-CF-min	1.380	1.700	0.905	0.645	0,753	0.881	
MR-CF-ave	1.439	1.779	*0.904*	0.633	0,745	0.880	
MR-CF-dim	**0.637**	**0.761**	**0.974**	**0.882**	**0,926**	**0.896**	

Table 5.7-1: *Evaluation results of the single and multiple collaborative filtering approaches as applied to ungrouped users and when 30and 50 profiles are formed.*

Depending on the depth of personalization that each application poses, UTARec system provides flexibility to examine every user individually. In **Figure 5.7-1**, for example, three evaluation metrics, MAE, Precision and Kendall's tau for a random user are plotted, vs. the number of clusters. It is noticed that for the specific user one rational decision would be to retrieve prediction values when 30 clusters are formed, where MAE has significantly decreased and Precision and Kendall' tau have almost reached their maximum levels. However, this decision may vary among users.

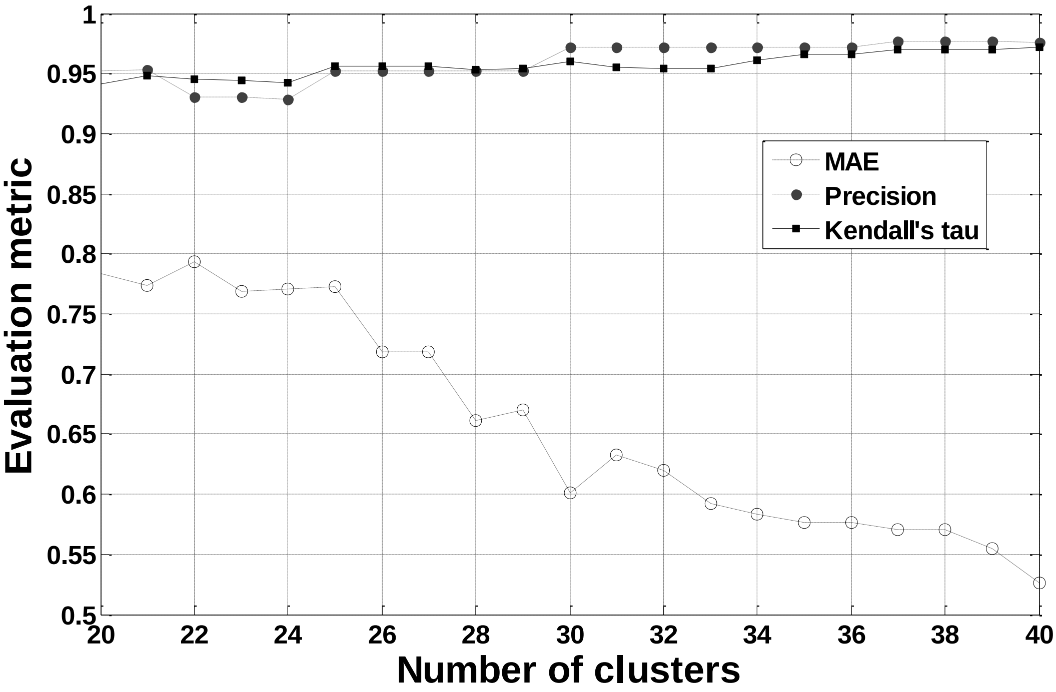


Figure 5.7-1: *MAE, Precision and Kendall's tau vs. number of clusters for a random user*

For the specific user, if we consider a 30 user profiling scheme, the relative improvement in MAE, Precision and Kendall's tau, compared to the ungrouped case, is 41%, 16% and 3%, respectively. The differences are attributed to the diverse nature of each measure and may vary across users.

5.8 Reference set size evaluation analysis

One possible attribute that is expected to affect recommendation accuracy is the reference set size. It is rational to expect that a larger reference set size used to model user's preferences would result to a more precise user modeling and thus to better recommendations. A question that arises at this point though is: "Is there an optimal number of reference set size?" To address this question the proposed methodological framework was applied to the third data set as discussed in 5.3.3, by varying each time the reference set size from 5 which is the minimum size, to 30. For every case the average Mean Absolute Error of all users was calculated. Although the third data set is significantly smaller than the first, it would be methodologically incorrect to compare the two different data sets. Although the evaluation metric that is used herein, the Mean Absolute Error is averaged over all users at any clustering step, and we wouldn't expected it to depend on the number of users, it is observed from **Figure 5.8-1** and **Figure 5.8-2** that not only absolute values of MAE change but also the slope by which the MAE curve degrades.

In **Figure 5.8-1** the average user MAE is plotted against the number of clusters for the first data set of 6078 users and when 5 alternatives are included in the reference set during the user modeling phase. In **Figure 5.8-2** the same measure is plotted for the second reference set this time, which includes 1716 users filtered out from the first data set, again modeled by preference information on five alternatives. The fact that average user MAE reaches lower values during the clustering process in the second data set, even if it starts from a higher value, can be easily attributed to the fact that, even if the same number of alternatives is included in the reference set, the users that are filtered out to form the second data set, are modeled with greater precision. The higher precision stems from the fact that when at least 10 alternatives are available to be used as reference, the system will choose those 5 that provide richer information, in other words it will avoid indifference relations and this will result in forming a more representative, in terms of preference information, reference set for every user.

From both **Figure 5.8-1** and **Figure 5.8-2** it is concluded that a number of 20 or 30 clusters may considered adequate for the specific data sets. A useful remark made from the analysis of these two figures is that in the first case of 6078 users, MAE curve appears less exponential than in the case of 1716 users. For example, in **Figure 5.8-1** the percentage degrease of average user MAE is 36% at 20 clusters and 44% at 30 clusters, whilst in **Figure 5.8-2** the same parameter decreased 51% when 20 clusters are formed and 58% when 30 clusters were formed. A possible, complementary explanation to that, apart from the difference in the user modeling quality discussed just above, is that smaller number of users requires fewer clusters to be assigned to.

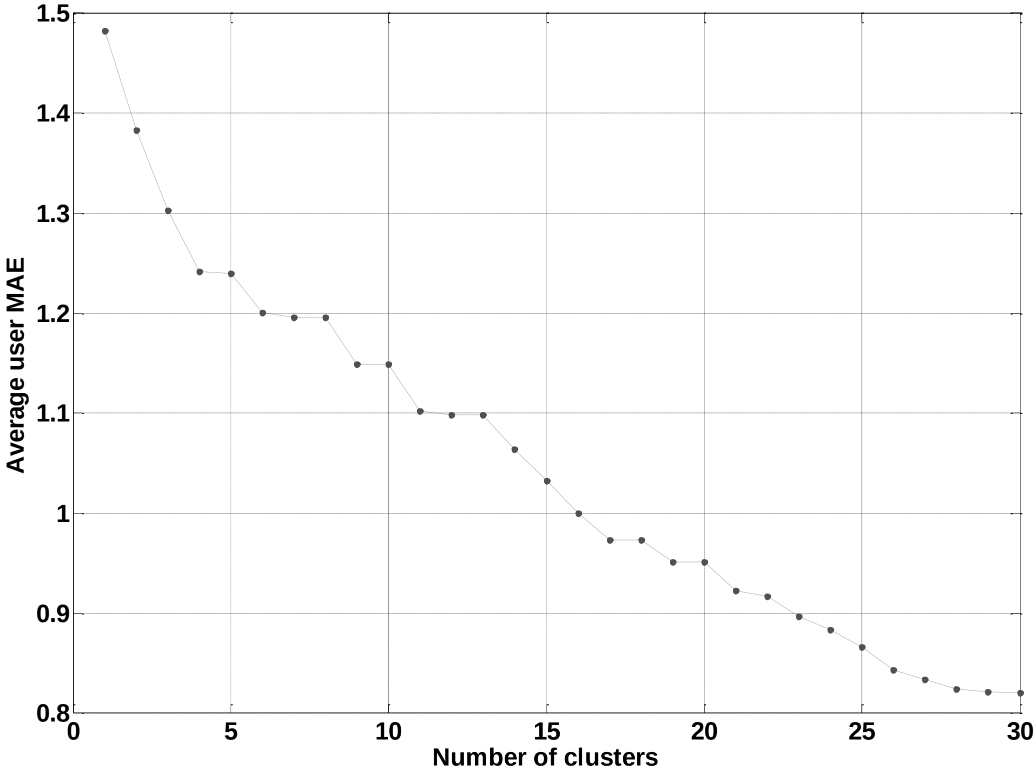


Figure 5.8-1: *Average user Mean Absolute Error versus the number of clusters for the first data set*

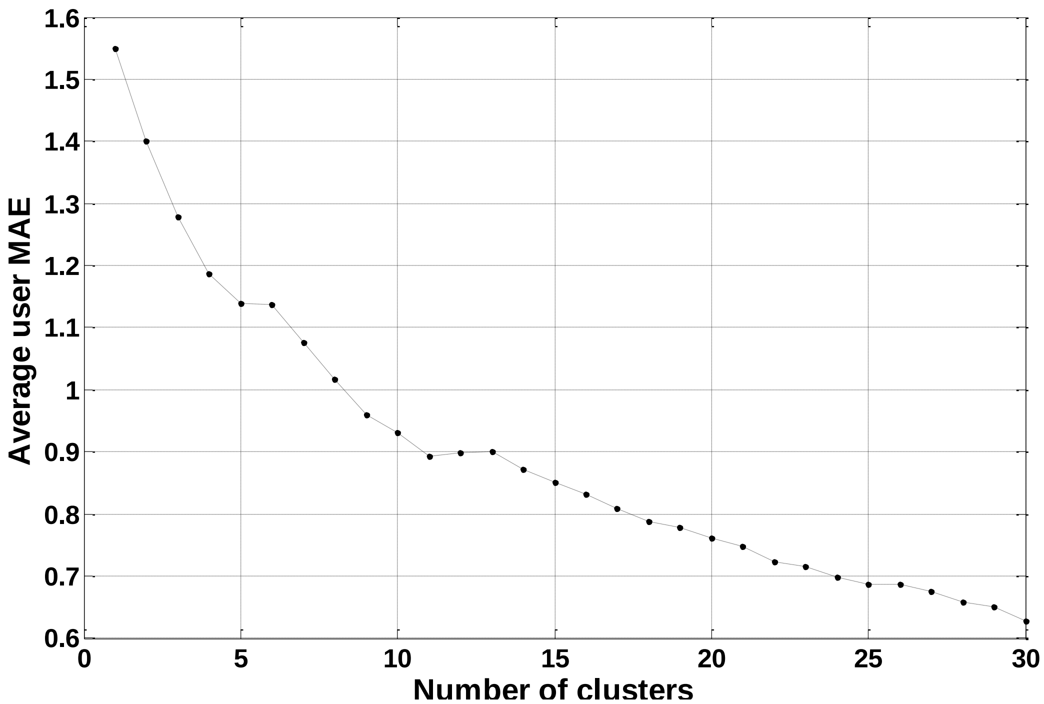


Figure 5.8-2: *Average user Mean Absolute Error versus the number of clusters for the second data set*

To further investigate this result average user MAE is also calculated for the third data set of 191 users and for different reference set sizes ranging from 5 to 30, without of course changing the number of users this time. Results are shown in **Figure 5.8-3.**

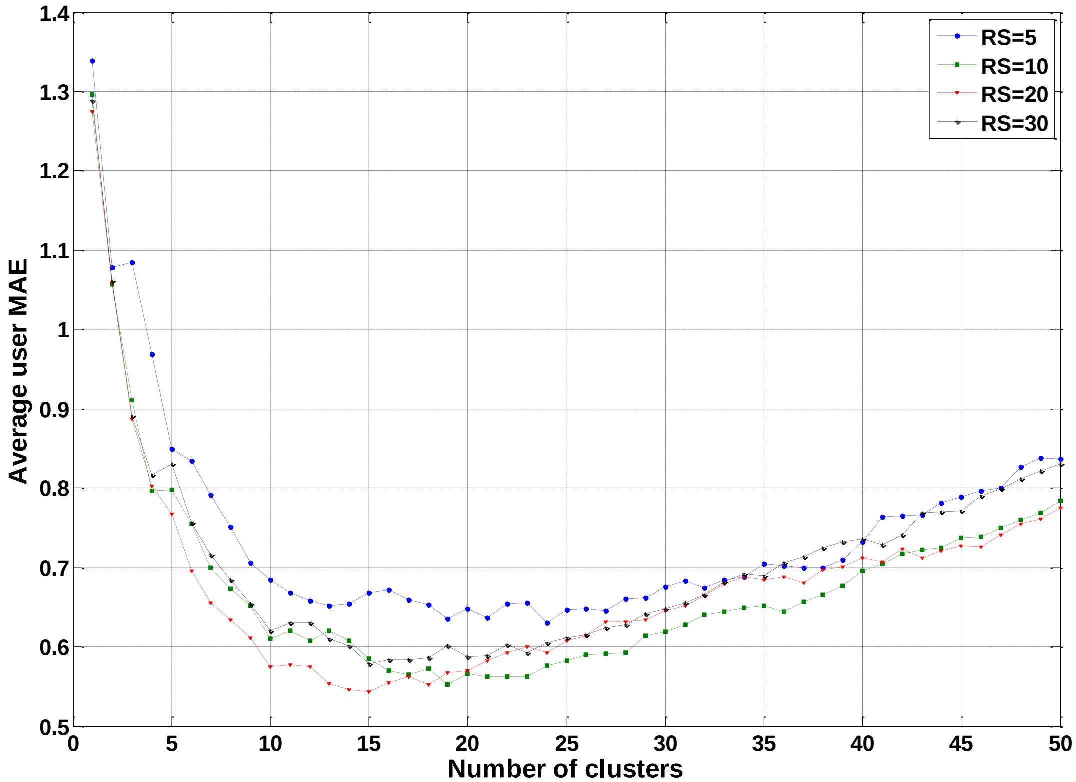


Figure 5.8-3: *Average user Mean Absolute Error versus the number of clusters for the third data set*

The aforementioned statement that the "optimal" number of clusters depends on the user modeling "quality" is more clearly proved in **Figure 5.8-3**. Two main conclusions can be drawn from **Figure 5.8-3** in combination with the two previous **Figure 5.8-1** and **Figure 5.8-2:**

1) The number of "optimal" user groups depends on user modeling precision.

2) A minor shift towards a smaller number of clusters is observed as the reference set size increases.

5.9 User profile group interpretation

As already stated, determining the optimal number of clusters is intrinsically ambiguous and thus, highly application-depended. Since both SSE and MAE are generally decreasing by the number of clusters formed, it is difficult to assign a stopping criterion for the number of clusters. One possible idea to address this problem is to interpret the multi-criteria user profiles according to their application interest (K. Lakiotaki and N. Matsatsinis 2009).

The cluster population can be considered as a potential parameter with commercial interest. Market segmentation is an important branch of marketing science (M. Wedel and W. Kamakura 2000). Market segmentation is a strategy that involves dividing a larger market into subsets of consumers who have common needs and applications for the goods and services offered in the market. These subgroups of consumers can be identified by a number of different demographic, geographic, psychographic, or behavioral characteristics depending on the purposes behind identifying the groups.

By adopting this idea of market segmentation to the proposed system, the individual behavior of 4 distinct groups is examined. This means that, for example, a movie with wide audience does not necessarily require an advanced clustering scheme, since many people would be interested to watch. On the other hand, a movie of specific interest requires a more sophisticated clustering to discover its appropriate recipients. For this reason, the frequency appearance of four distinct categorical groups of "movie taste" is studied. The first group concerns high populated clusters of "*massive taste*". Its size should be greater than the 50% of users. The second group concerns the "*broad taste*" movie group and is populated from 30% to 50% of users. The third group includes the so called "*selective taste*" users and is populated from 10% to 30% of users, while the last group of "*nonflexible taste*" is populated by a number of users less than 10%.

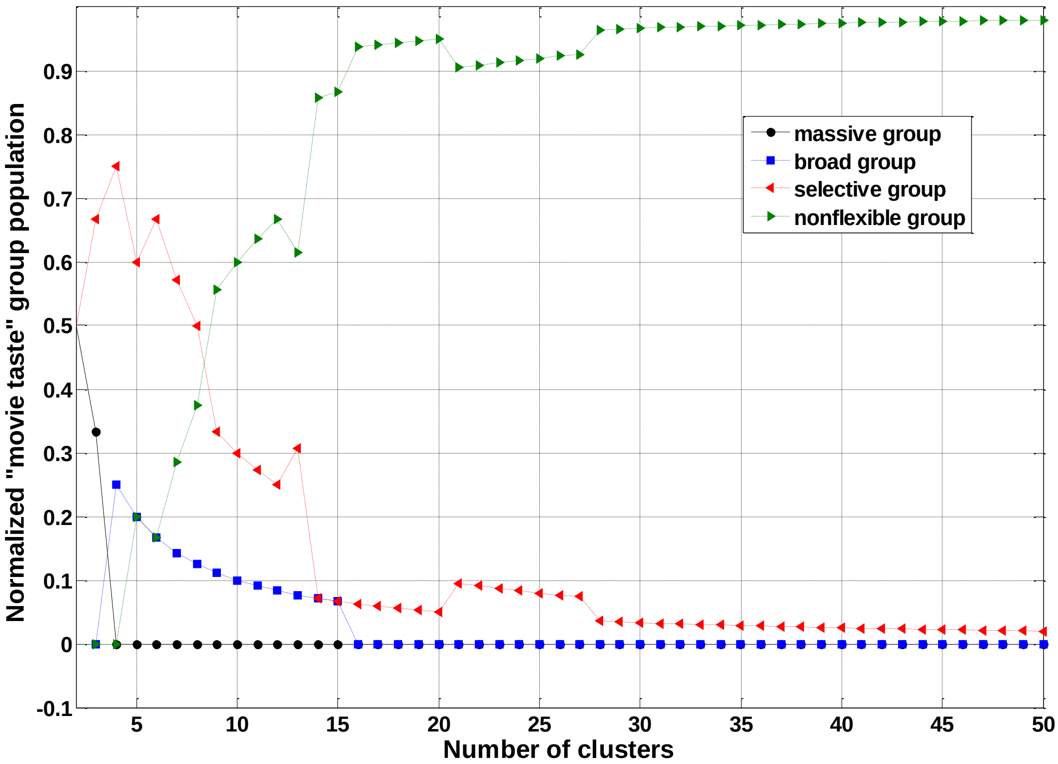


Figure 5.9-1: *Population of the various "movie taste" group vs. number of clusters for the first data set.*

As depicted in **Figure 5.9-1** in the case of recommending a "high chart" movie there is no need to create a large number of independent preference groups. It appears in **Figure 5.9-1** that for the "massive taste" group, when even 5 clusters or more are formed, the frequency by which this group appears reaches a plateau. For the "non flexible" group however, this plateau is not apparent for less than about 30 clusters.

These results seem to have no significant dependence on the reference set size for all four movie taste groups, which further validates the aforementioned outcome. Indicatively, in **Figure 5.9-2**, the evolution of the normalized "nonflexible" group population is shown vs. the number of clusters.

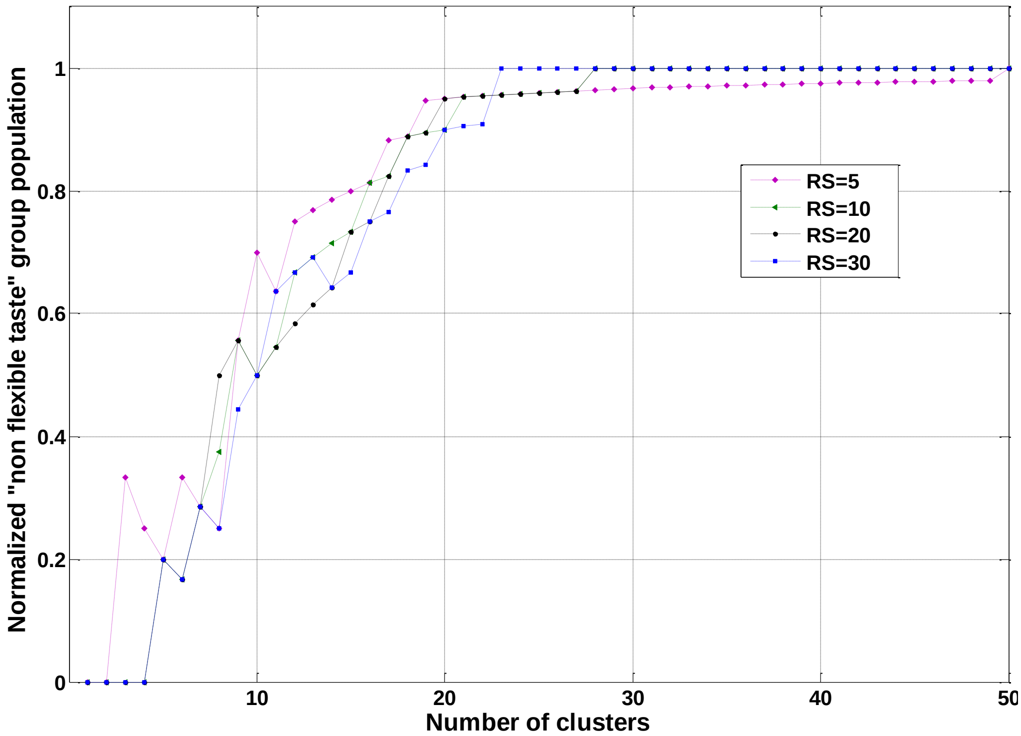


Figure 5.9-2: *Population of the "non flexible taste" group vs. number of clusters for the third data set for different reference set sizes.*

5.10 Conclusions

This chapter addressed the issue of UTARec evaluation. The design and implementation of a multi-criteria Recommender System imposes careful consideration. The novelty of incorporating unprecedented techniques for the Recommender Systems area in their widely exploited ideas, introduces the difficulty of examining any possible aspect of the resultant combination.

First, some preliminary results of the system were presented, mainly to unravel the issues that led to the improved last version of UTARec. Subsequently, the three data sets used in the evaluation analysis were statistically analyzed. The results regarding the second and third phase of the proposed methodology, the user modeling and the clustering phase respectively, were thereafter presented and important aspects of the systems such as the number of clusters, or the reference set size, were discussed. Specifically, it was shown that both the user similarity (calculated under a collaborative filtering philosophy), and the recommendation accuracy in terms of Mean Absolute

Errors from the real values of a test set, increase with the number of clusters, nevertheless, the majority of their growth is reached at about 30-50 clusters.

Furthermore, the trade off that appears at a certain number of clusters indicates that there is a lower limit point regarding the number of clusters, that when surmounted, recommendation accuracy is decreased. This decrease is less abrupt and in no case reversible to the initial trace followed to reach this lower limit.

Comparison analysis of UTARec's performance regarding other traditional methods employed in Recommender Systems, proved the higher, in terms of accuracy, recommendation performance of UTARec. System's prediction accuracy was compared not only to a single rating approach, to focus on the multi-criteria contribution, but also to other multi-criteria methodologies that are met throughout literature, to reveal the role of the multi-criteria user modeling and clustering methodology as proposed in this thesis.

The personalization flexibility of UTARec was also pointed out in section 5.9, via the user profile group interpretation. UTARec's level of personalization can be adjusted according to the movie profile, meaning the group of "movie taste" that the candidate for recommendation movie belongs to.

6 Concluding remarks

Contents

6.1 Summary and conclusions

This thesis studied Recommender Systems from a Multiple-Criteria Decision Analysis perspective. It was proved that, two different fields of research, the one of Recommender Systems, a recently developed field of Information Retrieval and the other of Multiple-Criteria Decision Analysis, a mature now field of Decision Science, share common goals and objectives and thus their merging, can be proved extremely dynamic and effective. Nevertheless, careful consideration must be taken to achieve maximum results.

Recommender Systems are being developed for about 20 years now and despite their exponential growth, they are still considered in their infancy from a research point of view. This means that yet, several aspects of these systems are to be explored.

Recommender Systems grew out of Information Retrieval, to overcome the natural consequence of the information age, which is called information overload and often leads users of Information Systems to despair and frustration. As a consequence, these systems try to filter out useful information for the respective user. To achieve that, accurate user modeling is undoubtedly considered their most important step and thus user profiling and modeling consist brotherly fields of Recommender Systems.

On the other hand, Multiple Criteria Analysis, is extensively studied in Operations Research due to the fact that real world problems are intrinsically multidimensional and many Multiple Criteria Analysis have nowadays developed and successfully applied to several managerial and other problems.

Multiple criteria Analysis may be considered as a set of methodologies that process several criteria simultaneously. This means that by default, an MCDA methodology assumes that multiple, often conflicting criteria are involved in a decision making process. To this end, it tries to model decision maker's value system by considering all the underlying attributes that lead to a specific decision.

It is advocated in this thesis that methodologies from the MCDA field can be proved helpful in solving common problems of Recommender Systems.

In particular, fully automated collaborative filtering Recommender Systems suffer from the so called "cold start" problem. This problem, either concerning new items or new users of the system, is apparent when insufficient information is gathered for this item or user. In UTARec, however, the cold start problem for new items is limited by the fact that, even if only once this item is rated, it enters the system and is ready to be recommended to all users that belong to the same group with the user that provided the initial rating. The way in which users are modeled under the framework of this thesis, enables a less vulnerable to cold start method, since a new user, is assigned in a group and automatically adopts properties of this group and a new item as soon as it is rated by one user it enters the system and is likely to be recommended to many users simultaneously.

Moreover, the data sparseness of traditional collaborative filtering Recommender Systems is limited in the case of a UTARec type Recommender

System, because neighbors in the latest case are defined due to a reference set evaluation and there is no need in discovering peers users with similar past behavior. As soon as initial preference information is provided the user can be assigned to a group and recommendations are immediately available.

An additional common problem of existing collaborative filtering systems is the unusual rater problem. This problem is attributed to the fact that some users exhibit unique preference behavior and to them a typical collaborative filtering system finds difficulty in recommending items, since it rarely identifies neighbors for this user. In a system designed on the proposed methodology, the initial reference set evaluation ensures the construction of a value system for this user and its assignment to a predefined group. Moreover, in case a user belongs to an underpopulated group the level of personalization can be adjusted and recommendation will be made at a high personalization level, meaning when several clusters will have been formed to ensure an adequate personalization status.

Last but not least, the major drawback of the second most popular type of Recommender Systems, the content based approach systems, is also not found in systems developed under the proposed approach, since no feature extraction occurs. The items to be recommended are not characterized at any point and the proposed approach is considered independent on item characteristics.

In summary, the main conclusions that can be drawn from the results of the UTARec analysis, which is considered as a multi-criteria movie Recommender System representing the proposed methodological framework, are gathered below:

- Multiple criteria user modeling increases prediction accuracy compared to single criterion modeling.
- User profile grouping based on multi-criteria user modeling improves recommendation accuracy independent on the notion of similarity used in the experiments of this thesis. Specifically, in all three multi-criteria, as well as in the single criterion notions of similarity, clustering was proved helpful.

- UTARec based systems limit some of the major shortcomings of existing Recommender Systems' methodologies, such as the cold start problem (either user or item cold start), the data sparseness, the unusual rater issue or the feature extraction dependence.
- The feedback option provides an alternative of consecutively improving recommendation accuracy, since more alternatives are available to be introduced to reference set, if necessary. A plateau seems to be reached when the reference set size exceeds 30.
- The option of adjusting the personalization level provides an additional feature of UTARec based systems, which endows these systems with personalization flexibility in order to achieve even higher recommendation scores.

6.2 Future aspects

The proposed multi-criteria user modeling methodology for designing and developing Recommender Systems is considered novel in the research field of these systems. Novelty, as already mentioned, requires careful consideration to examine all possible attributes that affect systems' performance. At the same time nevertheless, indicates several research directions under which potential researchers of this field may shift.

This thesis established an open research direction, that is, how multiple criteria ratings can be considered for improving Recommender Systems' effectiveness and functionality. Several aspects can be considered towards this direction, part of them mentioned below:

- The proposed methodological framework was presented and tested as a movie Recommender System. Undoubtedly, additional applications would establish the proposed methodology as a generic multi-criteria user modeling methodology for Recommender Systems and would obviously increase its range.
- In the user modeling phase of the discussed approach, the Disaggregation-Aggregation approach was chosen to stand for the multi-

criteria nature of the user modeling process. This approach was mainly chosen because its formulation best suits the requirements of a Recommender Systems, for instance the short user-system interaction. The extensive study, in combination with the numerous applications of the Disaggregation-Aggregation approach strongly encouraged its choice. Nevertheless, alternative multi-criteria approaches may prove even more helpful and appropriate for Recommender Systems. Further investigation towards existing or even modified or new multi-criteria methodologies is considered essential.

- The clustering process constituting the third phase of the proposed methodology was accomplished via the global k-means, a deterministic approach of the typical and highly popular k-means clustering algorithm. It would be especially interesting if alternative clustering algorithms that would probably result in different clustering schemes are studied.

- The final rating, according to which recommendations are made, is calculated by formula 4.2.4-7 that represents a weighted by the collaborative filtering similarity mean. Other mathematical formulas, for instance weighted by centroid distances, can also be tested to further examine the recommendation accuracy of such systems.

- Additionally, several techniques to fill the gaps of unknown ratings for the users can be used, to improve system's recommendation diversity. For example, all users that belong to the same group may be considered in the recommendation calculations, independent on whether these users have rated the candidate item, as long as their ratings are completed by a simple average of their past ratings for instance, or any other more sophisticated formula or method.

- Another important issue that has not been addressed in this thesis is the rating scale effect. Yahoo movies used a scale ranging from 1 to 13 to declare preference. For example Cosley et al.(D. Cosley, S. K. Lam*, et al.* 2003), empirically investigated the effect that several different rating scales have on user satisfaction and prediction accuracy in the area of Recommender Systems and showed that users rate highly on non zero and binary scales. In general, the choice of rating scales is a major concern in survey design. In this thesis, the rating scale for all data sets was a priori decided by Yahoo, leaving no choice in altering it, since this would introduce a bias. To reliably study this effect, experiments need to be conducted with the same users evaluating items on different rating scales. The source and format of the data set used in this thesis unfortunately left no room to study this effect.

7 Bibliography

1. A. M. Acilar and A. Arslan (2008). "A collaborative filtering method based on artificial immune network." *Expert Systems with Applications*, **36**, (4): 8324-8332
2. G. Adomavicius and Y. Kwon (2007). "New Recommendation Techniques for Multicriteria Rating Systems." *IEEE Intelligent Systems*, **22**, (3): 48-55.
3. G. Adomavicius and Y. O. Kwon (2007). "New Recommendation Techniques for Multicriteria Rating Systems." *IEEE Intelligent Systems*, **22**, (3): 48-55.
4. G. Adomavicius, R. Sankaranarayanan, S. Sen and A. Tuzhilin (2005). "Incorporating contextual information in recommender systems using a multidimensional approach." *ACM Transactions on Information Systems*, **23**, (1): 103 - 145.
5. G. Adomavicius and A. Tuzhilin (2005). "Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions." *IEEE Transactions on Knowledge and Data Engineering*, **17**, (6): 734-749.
6. A. Albadvia and M. Shahbazi (2009). "A hybrid recommendation technique based on product category attributes." *Expert Systems with Applications*, **36**, (9): 11480-11488.
7. R. B. Allen (1990). "User models: theory, method, and practice." *International Journal of Man-Machine Studies*, **32**, (5): 511-543
8. S. Amer-Yahia, A. Galland, J. Stoyanovich and C. Yu (2008). *From del.icio.us to x.qui.site: Recommendations in Social Tagging Sites*. SIGMOD, Vancouver, BC, Canada.
9. L. Ardissono, A. Goy, G. Petrone, M. Segnan and P. Torasso (2003). "Intrigue: Personalized Recommendation Of Tourist Attractions For Desktop And Handset Devices " *Applied Artificial Intelligence* **17**, (8-9): 687-714.
10. M. Balabanović and Y. Shoham (1997). "Fab: content-based, collaborative recommendation." *Communications of the ACM*, **40**, (3): 66-72.
11. D. Billsus, M. J. Pazzani and J. Chen (2000). *A learning agent for wireless news access*. 5th international conference on Intelligent user interfaces, New Orleans, Louisiana, United States.
12. D. Bouyssou, T. Marchant, M. Pirlot, A. Tsoukiàs and P. Vincke (2007). *Evaluation and Decision Models with Multiple Criteria: Stepping stones for the analyst* NY, USA, Springer.
13. R. I. Brafman, D. Heckerman and G. Shani (2003). *Recommendation as a Stochastic Sequential Decision Problem*. International Conference on Automated Planning & Scheduling Trento, Italy.

14. J. S. Breese, D. Heckerman and C. Kadie (1998). *Empirical Analysis of Predictive Algorithms for Collaborative Filtering*. Microsoft Research.

15. P. Brusilovsky, A. Kobsa and W. Nejdl, Eds. (2007). *The Adaptive Web: Methods and Strategies of Web Personalization*. Lecture Notes in Computer Science. Berlin Heidelberg, Springer-Verlag.

16. L. Buchanan and A. O'Connel (2006). "A brief history of Decision Making." *Harvard Business Review*.

17. R. Burke (2005). Hybrid Systems for Personalized Recommendations. *Intelligent Techniques for Web Personalization*. B. Mobasher and S. S. Anand. New York, Springer Verlag.

18. R. D. Burke, K. J. Hammond and B. C. Young (1997). "The FindMe Approach to Assisted Browsing." *IEEE Expert*, **12**: 32--40.

19. L. M. d. Campos, J. M. Fernández-Luna and J. F. Huete (2008). "A collaborative recommender system based on probabilistic inference from fuzzy observations." *Fuzzy Sets and Systems* **159** (12): 1554-1576

20. J. Canny (2002). *Collaborative filtering with privacy via factor analysis*. 25th annual international ACM SIGIR conference on Research and development in information retrieval, Tampere, Finland

21. K. Chapphannarungsri and S. Maneeroj (2009). *Combining Multiple Criteria and Multidimension for Movie Recommender System*. International MultiConference of Engineers and Computer Scientists, Hong Kong.

22. L. Chen (2008). *User Decision Improvement and Trust Building in Product Recommender Systems*.Lausanne, EPFL.

23. S. H. Choi, S. Kang and Y. J. Jeon (2006). "Personalized recommendation system based on product specification values." *Expert Systems with Applications*, **31**: 607–616.

24. W. W. Cohen (1995). *Fast Effective Rule Induction*. Twelfth International Conference on Machine Learning, Tahoe City, California, USA.

25. D. Cosley, S. K. Lam, I. Albert, J. A. Konstan and J. Riedl (2003). *Is Seeing Believing? How Recommender Interfaces Affect Users' Opinions*. SIGCHI conference on Human factors in computing systems Ft. Lauderdale, Florida, USA

26. B. J. Dahlen, J. A. Konstan, J. Herlocker, N. Good, A. Borchers and J. Riedl (1998). *Jump-starting movielens: User benefits of starting a collaborative filtering system with "dead data"*. University of Minnesota.

27. T. Fawcett (2003). *ROC Graphs: Notes and Practical Considerations for Data Mining Researchers*. HP Labs.

28. J. Figueira, S. Greco and M. Ehrgott (2005). *Multiple Criteria Decision Analysis: State of the Art Surveys*. Boston, Dordrecht, London, Springer Verlag.

29. J. Figueira, S. Greco and M. Ehrgott, Eds. (2005). *Multiple criteria decision analysis:State of the art surveys*. Boston, Springer.

30. G. Fischer (2001). "User Modeling in Human–Computer Interaction." *User Modeling and User-Adapted Interaction*, **11**, (1-2): 65-86.

31. E. Frias-Martinez, S. Y. Chen and X. Liu (2006). "Survey of Data Mining Approaches to User Modeling for Adaptive Hypermedia." *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, **36**, (6).

32. S. Gauch, M. Speretta, A. Chandramouli and A. Micarelli (2007). User Profiles for Personalized Information Access. *The Adaptive Web*. A. K. P. Brusilovsky, and W. Nejdl. Heidelberg, Springer-Verlag Berlin**:** 54 – 89.

33. P. W. Glimcher, C. Camerer, R. A. Poldrack and E. Fehr, Eds. (2008). *Neuroeconomics: Decision Making and the Brain* Elsevier Academic Press

34. D. Godoy and A. Amandi (2006). "Modeling user interests by conceptual clustering " *Information Systems*, **31**, (4-5): 247-265

35. D. Goldberg, D. Nichols, B. M. Oki and D. Terry (1992). "Using Collaborative Filtering to Weave an Information Tapestry." *Communications of the ACM*, **35**, (12): 61-70.

36. J. L. Herlocker, J. A. Konstan, L. G. Terveen and J. T. Riedl (2004). "Evaluating collaborative filtering recommender systems." *ACM Trans. Inf. Syst.*, **22**, (1): 5-53.

37. E. Jacquet-Lagreze and Y. Siskos (1982). "Assessing a set of additive utility functions for multicriteria decision-making, the UTA method " *European Journal of Operational Research*, **10**, (2): 151-164

38. E. Jacquet-Lagreze and Y. Siskos (2001). "Preference disaggregation: 20 years of MCDA experience." *European Journal of Operational Research*, **130**: 233-245.

39. A. Jameson (2004). *More Than the Sum of Its Members: Challenges for Group Recommender Systems*. working conference on Advanced visual interfaces, Gallipoli, Italy.

40. A. Johnson and N. Taatgen (2004). User Modeling. *The handbook of human factors in web design*. R. W. Proctor and K.-P. L. Vu, CRC.

41. S. Y. Jung, J.-H. Hong and T.-S. Kim (2005). "A Statistical Model for User Preference." *IEEE Transactions on Knowledge and Data Engineering*, **17**, (6): 834-843.

42. N. Karacapilidis and L. Hatzieleftheriou (2003). Exploiting Similarity Measures in Multi-criteria Based Recommendations. *E-Commerce and Web Technologies*, Springer Berlin / Heidelberg. **2738:** 424-434.

43. H. Kautz, B. Selman and M. Shah (1997). "Referral Web: combining social networks and collaborative filtering." *Communications of the ACM*, **40**, (3): 63-65.

44. S. Kazunari, H. Kenji and Y. Masatoshi (2004). *Adaptive web search based on user profile constructed without any effort from users*. 13th international conference on World Wide Web New York, NY, USA

45. R. Keeney and H. Raiffa (1993). *Decision with Multiple Objectives: Preference and Value Tradeoffs*. NY, Cambridge University Press.

46. L. Kelly and J. Dunnion (1999). *INVAID: an intelligent navigational AID far the world wide web*. IEE Colloquium on Microengineering in Optics and Optoelectronics.

47. J. Kennedy, R. C. Eberhart and Y. Shi, Eds. (2001). *Swarm Intelligence*. The Morgan Kaufmann Series in Evolutionary Computation.
48. D. Kim and B.-J. Yum (2005). "Collaborative filtering based on iterative principal component analysis." *Expert Systems with Applications*, **28**, (4): 823-830
49. T. Kliegr (2009). *UTA - NM: Explaining Stated Preferences with Additive Non-Monotonic Utility Functions*. ECML/PKDD-09 Workshop on Preference Learning Bled, Slovenia.
50. A. Kobsa (2001). "Generic User Modeling Systems." *User Modeling and User-Adapted Interaction*, **11**: 49^63.
51. J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon and J. Riedl (1997). "GroupLens: applying collaborative filtering to Usenet news." *Communications of the ACM*, **40**, (3): 77-87.
52. J. A. Konstan, J. Riedl, A. Borchers and J. L. Herlocker (1998). *Recommender Systems: A GroupLens Perspective*.
53. Y. Koren (2008). *Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model*. 14th ACM SIGKDD conference on Knowledge Discovery & Data Mining, Las Vegas, Nevada, USA, August 24–27.
54. Y. Koren (2009). *Collaborative Filtering with Temporal Dynamics*. 15th ACM SIGKDD conference on Knowledge Discovery and Data Mining, Paris, France.
55. B. Krulwich (1997). "LIFESTYLE FINDER Intelligent User Profiling Using Large-Scale Demographic Data." *AI Magazine*, **18**, (2).
56. K. Lakiotaki, P. Delias, V. Sakkalis and N. F. Matsatsinis (2009). "User profiling for Recommender Systems: The role of Utility Functions." *Operational Research: An International Journal*, **9**, (1): 3-16.
57. K. Lakiotaki and N. Matsatsinis (2009). *Analyzing User Modeling in a Multi-Criteria Movie Recommender System*. ACM Recommender Systems 2009, NY, USA.
58. K. Lakiotaki, N. Matsatsinis and AlexisTsoukiàs "Multi-Criteria User Modeling in Recommender Systems." *IEEE Intelligent Systems*.
59. K. Lakiotaki, S. Tsafarakis and N. Matsatsinis (2008). *UTA-Rec: a recommender system based on multiple criteria analysis*. ACM conference on Recommender systems, Lausanne, Switzerland.
60. T. Lampinen, M. Laurikkala, H. Koivisto and T. Honkanen (2005). Profiling Network Applications with Fuzzy C-means and Self-Organizing Maps *Classification and Clustering for Knowledge Discovery*.
61. D. T. Larose (2005). *Discovering Knowledge in Data: An Introduction to Data Mining.* New Jersey, John Wiley & Sons.
62. H. Lieberman (1995). *Letizia: An Agent That Assists Web Browsing*. Fourteenth International Joint Conference on Artificial Intelligence, Montreal, Quebec, Canada.
63. H. Lieberman, N. W. V. Dyke and A. S. Vivacqua (1998). *Let's browse: a collaborative Web browsing agent*. 4th international conference on Intelligent user interfaces, Los Angeles, California, United States

64. A. Likas, N. Vlassis and J. Verbeek (2003). "The global k-means algorithm." *Pattern Recognition*, **36**, (2): 451-461.

65. W. Lin and S. A. Alvarez (2002). "Efficient Adaptive-Support Association Rule Mining for Recommender Systems." *Data Mining and Knowledge Discovery*, **6**: 83--105.

66. F. Lorenzi, D. S. d. Santos and A. L. C. Bazzan (2005). *Case-based recommender system inspired by social insects*. XXV Congresso da Sociedate Braseilera de Computacao, São Leopoldo, Brazil.

67. H. Lyer (1998). *Electronic resources: use and user behavior*, Haworth Press.

68. J. B. MacQueen (1967). *Some Methods for classification and Analysis of Multivariate Observations*. 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley.

69. N. Manouselis and C. Costopoulou (2007). "Analysis and Classification of Multi-Criteria Recommender Systems." *World Wide Web: Internet and Web Information Systems*, **10**, (4): 415-441.

70. N. Matsatsinis, K. Lakiotaki and P. Delias (2007). *A System based on Multiple Criteria Analysis for Scientific Paper Recommendation*. 11th Panhellenic Conference on Informatics (PCI 2007), Patras, Greece, 18-20 May

71. K. McCarth, M. Salamó, L. Coyle, Lorraine McGinty, B. Smyth and P. Nixon (2006). *Group Recommender Systems: A Critiquing Based Approach*. 11th international conference on Intelligent user interfaces, Sydney, Australia.

72. D. W. McDonald (2003). "Ubiquitous Recommendation Systems." *Computer*, **36**, (10).

73. S. M. McNee, J. Riedl and J. A. Konstan (2006). *Making Recommendations Better: An Analytic Model for Human-Recommender Interaction*. ACM Conference on Human Factors in Computing Systems (CHI 2006), Montreal, Canada.

74. S. E. Middleton, D. D. Roure and N. R. Shadbolt (2004). Ontology-based Recommender Systems. *Handbook on Ontologies*. S. Staab and R. Studer, Springer**:** 577-498.

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

76. B. N. Miller, I. Albert, S. K. Lam, J. A. Konstan and J. Riedl (2003). *MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System*. International Conference on Intelligent User Interfaces, Miami, Florida, USA

77. M. Montaner, B. López and J. L. D. L. Rosa (2003). "A Taxonomy of Recommender Agents on the Internet." *Artificial Intelligence Review*, **19**, (4): 285 - 330.

78. O. Nasraoui, H. Frigui, A. Joshi and R. Krishnapuram (1999). *Mining Web Access Logs Using Relational Competitive Fuzzy Clustering*. Eight International Fuzzy Systems Association World Congress, Taipei, Taiwan.

79. N. Negroponte (1996). *Being Digital*, Vintage.

80. M. O'Connor, D. Cosley, J. A. Konstan and J. Riedl (2001). *PolyLens: A Recommender System for Groups of Users*. Europeon Conference on Computer Supported Co-Operative Work, Bonn, Germany.

81. G. Paliouras, V. Karkaletisis, C. Papatheodorou and C. D. Spyropoulos (1999). *Exploiting learning techniques for the acquisition of user stereotypes and communities*. seventh international conference on User modeling Banff, Canada

82. M. P. Papazoglou (2001). "Agent-oriented technology in support of e-business." *Communications of the ACM* **44**, (4): 71 - 77

83. M. Pazzani, J. Muramatsu and D. Billsus (1996). *Syskill & Webert: Identifying interesting web sites* Thirteenth National Conference on Artificial Intelligence, Portland, Oregon.

84. M. J. Pazzani and D. Billsus (2007). Content-Based Recommendation Systems *The Adaptive Web*. P. Brusilovsky, A. Kobsa and W. Nejdl. Germany, Springer Berlin / Heidelberg. **4321:** 325-341.

85. P. Perny and J. D. Zucker (2001). "Preference-based Search and Machine Learning for Collaborative Filtering: the "Film-Conseil" Movie Recommender System." *Inform. Interact. Intell.*, **1**, (1): 9-48.

86. S. Perugini, M. A. Gonzalves and E. A. Fox (2004). "Recommender Systems Research: A Connection-Centric Survey." *Journal of Intelligent Information Systems*, **23**, (2): 107-143.

87. P. Pu, L. Chen and P. Kumar (2008). "Evaluating product search and recommender systems for E-commerce environments." *Electronic Commerce Research*, **8**, (1-2): 1-27.

88. P. Pu and B. Faltings (2000). *Enriching buyers' experiences: the SmartClient approach*. Conference on Human factors in computing systems (SIGCHI) The Hague, The Netherlands.

89. J. R. Quinlan (1992). *C4.5: Programs for Machine Learning* San Francisco, CA, USA Morgan Kaufmann.

90. R. J. Quinlan (1986). "Induction of Decision Trees." *Machine Learning 1*, **1**: 81-106.

91. K. N. Rao and V.G.Talwar (2008). "Application Domain and Functional Classification of Recommender Systems—A Survey." *Journal of Library & Information Technology*, **28**, (3): 17-35.

92. J. Reilly, K. McCarthy, L. McGinty and B. Smyth (2005). "Incremental critiquing." *Knowledge-Based Systems* **18**: 143–151.

93. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom and J. Riedl (1994). *GroupLens: an open architecture for collaborative filtering of netnews*. ACM conference on Computer supported cooperative work, Chapel Hill, North Carolina, United States

94. P. Resnick and H. R. Varian (1997). "Recommender systems." *Communications of the ACM*, **40**, (3): 56-58.

95. F. Ricci and Q. N. Nguyen (2007). "Acquiring and Revising Preferences in a Critique-Based Mobile Recommender System." *IEEE Intelligent Systems*, **22**, (3).

96. E. Rich (1983). "Users are individuals:- individualizing user models." *International Journal of Man-Machine Studies*, **18**: 199-214.

97. J. Riedl, J. Konstan and E. Vrooman (2002). *Word of Mouse: The Marketing Power of Collaborative Filtering.* NY, Business Plus.

98. F. L. Roux, E. Ranjeet, V. Ghai, Y. Gao and J. Lu *A Course Recommender System Using Multiple Criteria Decision Making Method*. International Conference on Intelligent Systems and Knowledge Engineering, Chengdu, China.

99. B. Roy (1985). "Méthodologie Multicritère d'Aide à la Décision." *Economica*.

100. J. Rucker and M. J. Polanco (1997). "Siteseer: personalized navigation for the Web." *Communications of the ACM*, **40**, (3): 73-76.

101. N. Sahoo, R. Krishnan, G. Duncan and J. P. Callan "Collaborative Filtering with Multi-component Rating for Recommender Systems."

102. G. Salton (1971). *The SMART Retrieval System - Experiments in Automatic Document Processing*, Prentice Hall.

103. B. Sarwar, G. Karypis, J. Konstan and J. Riedl (2002). *Incremental SVD-Based Algorithms for Highly Scalable Recommender Systems*. 5th International Conference on Computer and Information Technology, Dhaka, Bangladesh.

104. J. B. Schafer, D. Frankowski, J. Herlocker and S. Sen (2007). Collaborative Filtering Recommender Systems. *The Adaptive Web: Methods and Strategies of Web Personalization*. P. Brusilovsky, A. Kobsa and W. Nejdl, Springer.

105. J. B. Schafer, J. Konstan and J. Riedl (1999). *Recommender Systems in E-Commerce*. 1st ACM conference on Electronic commerce, Denver, Colorado, United States

106. S. Sen, J. Vig and J. Riedl (2009). *Tagommenders: Connecting Users to Items through Tags*. International World Wide Web Conference, Madrid, Spain.

107. G. Shani, D. Heckerman and R. I. Brafman (2005). "An MDP-Based Recommender System." *Journal of Machine Learning Research*, **6**: 1265–1295.

108. U. Shardanand and P. Maes (1995). *Social Information Filtering: Algorithms for Automating "Word of Mouth"*. the 95' Conference on Human Factors in Computing Systems.

109. E. Shifroni and B. Shanon (1992). "Interactive user modeling: An integrative explicit-implicit approach." *User Modeling and User-Adapted Interaction*, **2**, (4): 331-365.

110. H. A. Simon (1977). *The New Science of Management Decision*, Prentice Hall PTR.

111. Y. Siskos, E. Grigoroudis and N. Matsatsinis (2005). UTA Methods. *Multiple criteria decision analysis:State of the art surveys*. J. Figueira, S. Greco and M. Ehrgott. Boston, Springer**:** 297-344.

112. B. Smyth (2007). Case-Based Recommendation. *The Adaptive Web: Methods and Strategies of Web Personalization*. P. Brusilovsky, A. Kobsa and W. Nejdl.

113. S. Stewart and J. Davies (1997). *User Profiling Techniques: A Critical Review*. BCS-IRSG Annual Colloquium on IR Research.

114. P.-N. Tan, M. Steinbach and V. Kumar (2006). Cluster analysis: Basic Concepts and Algorithms. *Introduction to data mining*, Addison-Wesley.

115. W.-G. Teng and H.-H. Lee (2007). "Collaborative Recommendation with Multi-Criteria Ratings." *Journal of Computers*, **17**, (4).

116. L. Terveen and W. Hill (2001). Beyond Recommender Systems: Helping People Help Each Other *HCI In The New Millennium*. J. Carroll, Addison-Wesley.

117. L. Terveen, W. Hill, B. Amento, D. McDonald and J. Creter (1997). "PHOAKS: a system for sharing recommendations." *Communications of the ACM*, **40**, (3): 59-62.

118. S. Tsafarakis, K. Lakiotaki and N. Matsatsinis Applications of MCDA in Marketing and e-Commerce. *Handbook of Multicriteria Analysis*. C. Zopounidis and P. Pardalos, Springer.

119. A. Tsoukiàs (2007). "On the concept of decision aiding process: an operational perspective." *Annals of Operations Research*, **154**, (1): 3-27.

120. P. Viappiani, P. Pu and B. Faltings (2007). *Conversational Recommenders with Adaptive Suggestions*. Recommender Systems (RecSys), Minneapolis, Minnesota, USA.

121. M. Wedel and W. Kamakura (2000). *Market Segmantation: Conceptual and Methodological Foundations*.

122. S. K. M. Wong and C. J. Butz (2000). "A Bayesian approach to user profiling in information retrieval,." *Technology Letters*, **4**, (1): 50–56.

123. G. Xu, Y. Zhang and X. Zhou (2005). Towards User Profiling for Web Recommendation *AI 2005: Advances in Artificial Intelligence*. S. Zhang and R. Jarvis, Springer Berlin / Heidelberg**:** 415-424.

124. Y. Xun and L. Quan-zhong (2007). *Immune-Inspired Collaborative Filtering Technology for Rating-Based Recommendation System*. IFIP International Conference on Network and Parallel Computing Workshops Liaoning, China.

125. J. Zaslow (2002). "If TiVo Thinks You Are Gay, Here's How To Set It Straight " *The Wall Street Journal*, (sect. A): 1.

126. T. Zhang and V. S. Iyengar (2002). "Recommender Systems Using Linear Classifiers." *Journal of Machine Learning Research*, **2**: 313-334.

127. Y. Zhang, Y. Zhuang, J. Wu and L. Zhang (2009). "Applying probabilistic latent semantic analysis to multi-criteria recommender system." *AI Communications*, **22**, (2): 97–107.

128. A. Zimdars, D. M. Chickering and C. Meek (2001). *Using Temporal Data for Making Recommendations*. 17th Conference in Uncertainty in Artificial Intelligence Seattle, WA.