CITY UNIVERSITY
LONDON

City University London

School of Engineering and Mathematical Sciences

A Dynamic Multi-Algorithm Collaborative-Filtering System

PhD Thesis

Supervisior: Dr. Muttukrishnan Rajarajan

Supervisior: Prof. Dr. Rudolf Jäger

Second Supervisior: Dr. Veselin Rakocevic

by

Dipl.-Ing. (FH) Christian Überall

September 2012

Table of Contents

List of Tables

List of Figures

List of Abbreviations

API	Application Programming Interface
AVATAR	**A**d**VA**nced **T**elematic search of **A**udiovisual contents by semantic **R**easoning
CPU	Central Processing Unit
DOI	Degree of Interest
DVB	Digital Video Broadcast
DVB-C	DVB-Cable
DVB-S	DVB-Satellite
DVB-T	DVB-Terrestrial
EIT	Event Information Table
EPG	Electronic Program Guide
ETSI	European Telecommunications Standards Institute
FTP	File Transfer Protocol
GUI	Graphical User Interface
HDD	Hard Disc Drive
IF	Information Filtering

KNN	K-Nearest Neighbour
MAE	Mean Absolute Error
MSE	Mean Square Error
OWL	Web Ontology Language
PC	Personal Computer
PHP	Hypertext Preprocessor
PPG	Personal Program Guide
QoS	Quality of Service
RAM	Random Access Memory
RI	Recommendation Index
RMSE	Root Mean Square Error
ROM	Read-Only Memory
SI	Service Information
SotA	State-of-the-Art
STB	Set-Top-Box
SVD	Singular Value Decomposition
THM	Technische Hochschule Mittelhessen - University of Applied Sciences
TV	Television
URL	Uniform Resource Locator
USB	Universal Serial Bus
VCR	Video Cassette Recording
WWW	World Wide Web
XML	Extensible Markup Language

Acknowledgements

First of all I want to thank my wife and my family, who always supported me during the whole process. Secondly I want to thank Muttukrishnan Rajarajan for the great supervision and the support as well. Rudolf Jäger offered me the opportunity to do my research work. Without his commitment I would not have been able to do my PhD research work. Veselin Rakocevic also assisted me if I had any questions. Last but not least I want to thank Christopher Köhnen, who always helped me if I had any problems. All these persons really helped me to do my research work successfully and I am highly grateful for their support.

Declaration

I declare that the University Librarian is allowed to copy the presented thesis in whole or in part without referencing me. The permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

Date	Christian Überall

Abstract

Nowadays users have access to an immense number of media content. They are able to consume thousands of Television (TV) channels and millions of video clips from online portals like YouTube. Due to the immense number of available content, users can have the problem to find content of interest. This problem can be solved by recommendation systems. For example, recommendation systems can be used to create recommendations which fit to the preferences of users.

Recommendation systems can use two different approaches for the creation of recommendations. They can take content-based and/or collaborative-filtering techniques into account. Content-based filtering techniques use information, the so-called metadata, that describe the content in more detail. Collaborative-filtering techniques calculate similarities e.g., between users. All users are included in a dataset, the so-called community. Generally the number of user profiles within the community is quite large. Examples of such huge communities are Amazon, Netflix, MovieLens, and LastFM. The community which includes the user profiles is used to create a user-item matrix. This user-item matrix contains the preferences from users on items e.g., movies, genres, book titles, and so forth.

The quality of the recommendations depends on the accuracy of the predictions. As mentioned above, collaborative-filtering techniques calculate similarities e.g., between users. These similarities can be used to calculate predictions for an entry within the user-item matrix. If the predictions are close or equal to the preferences of a user, the used collaborative-filtering technique predicts accurately.

Generally recommendation systems only use one single collaborative-filtering algorithm for the similarity calculation. The research work of this thesis proves that a dynamic selection of the most accurate filtering algorithm by considering more algorithms is able to increase the accuracy of the predictions significantly.

In order to increase the accuracy of predictions, this thesis presents a dynamic multi-algorithm collaborative-filtering system which creates recommendations for video content, such as movies or genres. This system is able to find the most accurate filtering algorithm by considering the k-nearest neighbours. These neighbours are selected by identifying the most similar users or items e.g., movies. Besides the dynamic selection, this thesis presents newly developed collaborative-filtering algorithms which are able to overcome researched weaknesses of state-of-the-art algorithms.

The evaluation of the proposed system considers a huge dataset from MovieLens and a small dataset from an undertaken survey. The consideration of a huge and a small dataset shall prove that the system can be used in both cases.

The results of this thesis show that the proposed system is able to decrease the error rate significantly compared to existing approaches.

Chapter 1

Introduction

In the 1990s the available information and entertainment technologies increased exponentially. Users have access to hundreds of TV channels, thousands of videos from online portals, millions of books, news, web pages, images, and CDs on the World Wide Web (WWW) [1,2]. Due to this immense amount of available content, users are overloaded with information [3]. In order to filter the available information, recommendation systems have become important [2]. Recommendation systems use the Information Filtering (IF) technique to present and recommend items, such as books, movies, images, and so forth, that could be interesting to individual users [1]. An "item" is a term that is used to donate what a recommendation systems recommends to users [4]. They filter the content and create recommendations by the usage of different approaches.

Basically a recommendation system can create non-personalized and personalized recommendations.

A recommendation system which creates non-personalized recommendations offers identical recommendations to each user. These kind of recommendation systems creates recommendations for e.g., products to customers based on the feedback of other customers. The recommendations are independent of the individual customer. Each customer gets the same recommendation [5]. Typical examples for those kind of recommendations are the top ten selection of books, CD etc.

A personalized recommendation system creates the recommendations based on preferences which represents the likings on e.g., a specific movie, book, genre, and so forth. The preferences can be created in an implicit and in an explicit manner. A personalized recommendation system uses either the content-based approach, the collaborative-filtering approach, or a combination of these two approaches. The content-based approach uses the metadata which describes the content in more detail. For example, if a user prefers to watch documentaries, a recommendation system can personalize the recommendations by taking this preference into account. The collaborative-filtering approach considers the preferences from several users for the creation of recommendations. The first system which used this approach was introduced by Goldberg *et al.* [6].

However, this thesis focuses on personalized recommendation systems by the usage of collaborative-filtering techniques. The following paragraphs therefore describe the collaborative-filtering approach in more detail.

Recommendations which use the dataset from a community can be created by the usage of collaborative-filtering techniques [7]. A user-item matrix that contains the ratings from the users on items (e.g., movies) represents the community. The ratings represent the preference in an item and is quite often within a range [0;5], where zero represents no interest and five definite interest. Generally this community contains a huge number of ratings. Examples of huge communities are Amazon, Netflix, MovieLens, and LastFM. Recommendations are based on similarities between users or items. These similarities can be calculated by the usage of collaborative-filtering algorithms, such as the *Pearson-r Correlation*, the *Spearman Rank Correlation*, the *Cosine Similarity*, or the *Adjusted Cosine Similarity*. The similarities between users or items can be used to calculate predictions [8]. The result of the prediction calculation predicts a rating within the user-item matrix. With this technique entries of the user-item matrix can be predicted which are not rated yet. The calculation of the predictions can be performed by taking the *Weighted Sum* [9–12] approach into account. The predictions' accuracy can be exploited by the usage of the Mean Absolute Error (MAE). Besides the MAE [10, 11, 13, 14], the Mean Square Error (MSE) or the Root Mean Square Error (RMSE) [12, 15–19] are also able to exploit the accuracy of the predictions. Besides the calculation of the predictions, the similarities can also be used to find the k-nearest neighbours. The k-nearest neighbours include users or items that are quite similar to the active user or active item [12].

The similarity calculation is normally realized by the usage of one single collaborative-filtering algorithm. The evaluations of this thesis prove that the most accurate algorithm is strongly connected to the active user/item and its neighbourhood, which is represented by the k-nearest neighbours. The results of the evaluation show that the most accurate collaborative-filtering algorithm can not be reduced to a single one. In addition to these results, the thesis proves that a dynamic selection of the most accurate collaborative-filtering algorithm can significantly reduce the error rates, such as the MAE, MSE, or the RMSE. This reduction of the error rates improves the predictions' accuracy.

Besides these improvements, the presented thesis also shows up some investigated weaknesses of State-of-the-Art (SotA) collaborative-filtering algorithms. This thesis presents newly developed collaborative-filtering algorithms that overcome researched weaknesses of the *Pearson-r Correlation*, the *Spearman Rank Correlation*, the *Cosine Similarity*, and the *Adjusted Cosine Similarity*. The evaluation of this thesis proves the usefulness of the newly developed algorithms.

The main contribution is a researched and developed dynamic multi-algorithm collaborative-filtering system. It includes the mentioned SotA algorithms and newly researched and developed algorithms that are able to overcome researched weaknesses. The proposed system finds the most accurate filtering algorithm by taking the active user or active item and its k-nearest neighbours into account. The finding of the most accurate collaborative-filtering algorithm is realized by the exploiting of the error rates, such as the MAE, the MSE, and the RMSE. The algorithm which produces the lowest error rate will be proposed for further calculations.

In order to prove the usefulness of the proposed system, this thesis considers two datasets. The first dataset from MovieLens represents a huge community. This dataset includes ratings from 943 users and 1682 items (movies). The evaluation of this thesis takes this dataset into account and proves that the error rates are significantly lower than the error rates from existing approaches, which also use this dataset. In addition to the dataset from MovieLens, this thesis also considers a dataset from a survey. This dataset represents a small community. The user-item matrix which contains the ratings is presented in Table 3.1. The survey was undertaken at the Technische Hochschule Mittelhessen - University of Applied Sciences (THM). Each respondent was asked to rate genres that are specified by a European Telecommunications Standards Institute (ETSI) standard for Digital Video Broadcast (DVB) Service Information [20]. This ETSI standard specifies twelve main genres, that are delivered within the DVB Transport Stream. The setting of the ratings could be realized by using an interface [21, 22]. The respondents were able to set their preferences by setting stars. Five stars represent definite interest in the selected genre and zero stars represent no interest in the selected genre. These settings were saved as the explicit user profile.

The following sections present the problem statement and the motivation of this thesis. Additionally they present the objectives of this thesis and the report contribution. Finally the organization of this thesis is described.

1.1 Problem Statements and Motivation

Nowadays users have access to an immense amount of media content. The classic broadcast TV offers thousands of TV channels. Novel generations of TV devices are also able to access media content which is available via the Internet. A study which was undertaken by BITCOM [23] in 2011 showed that every second TV sold in Germany is Internet capable. Thus, 5 million TVs with a built-in web connection were sold in 2011. That is almost a tenfold increase within two years. All older flat panels can be upgraded for Internet reception using hybrid set-top boxes. These set-top boxes are also available with an integrated hard drive for recording programs, so Video Cassette Recording (VCR) becomes redundant. Almost every second German (46 percent) wants to connect their TV to the Internet to watch web content on their TV device. These figures are the results of a representative survey in the study "The Future of Consumer Electronics". Especially the younger generation longs for the TV-web. 60 percent of Germans between 14 and 26 years want to have a TV which is connected to the Internet. 74 percent of young Americans and 77 percent of young Britons want to have an Internet TV. Besides these aspects, nowadays clients are connected within a home environment through the network. An example of a home environment is presented in Section 3.1. The connection of the linear DVB content and the non-linear content from the Internet offers an immense amount of media content. DVB-Satellite (DVB-S) [24] offers more than 1000 channels, DVB-Cable (DVB-C) [25] offers more than 200 channels, and DVB-Terrestrial (DVB-T) [26] more than 50 channels. Users are overloaded with information and they have the problem of finding content of interest in less time [27–30]. In order to overcome this problem, recommendation systems can be used to find content of interest.

Collaborative-filtering techniques can be used to generate recommendations by using data from a community [31–40]. Existing approaches use data from huge communities such as MovieLens, Netflix, or LastFM. Typically recommendation systems take one collaborative-filtering algorithm into account. Research studies of the presented thesis prove that the most accurate algorithm is strongly connected to the given user-item matrix, active user/item, and its neighbourhood. An algorithm, which performs the best results by considering an user-item matrix can provide the worst results by using another user-item matrix. Due to these facts the main challenge of this thesis is the research and development of a recommendation system that selects the most accurate algorithm which is strongly connected to the active user or item. Another disadvantage of existing approaches is the limitation of the evaluation by considering small datasets. As mentioned above, collaborative-filtering systems normally use a huge dataset. This thesis will also address user-item matrices which contain only a small number of users or items.

The quality of recommendation systems is strongly connected to the accuracy of the predictions. Several recommendation systems are evaluated by the calculation of error rates [7, 10, 11, 14–18, 40]. For example, Netflix, a movie recommender, ran a competition between 2007 and 2009. The aim of this competition was the accuracy improvement of predictions by the usage of collaborative-filtering techniques. Challengers were asked to reduce the error rate of these predictions. They had to reduce the RMSE by about ten percent. Challengers who were able to produce the lowest RMSE, won 1,000,000 USD.

The main task of this thesis is the improvement of the predictions' accuracy by reducing the error rate and considering small and large datasets. These objectives will be described below.

1.2 Objectives of the Thesis

Since the quality of a recommendation system is strongly connected to the accuracy of the predictions, this thesis will focus on the accuracy improvement by the usage of collaborative-filtering techniques. The proposed system shall be able to reduce the error rate significantly compared to existing approaches. In addition, the proposed system shall be able to consider small and large datasets. The small dataset shall represent the preferences of, for example, a family, a block of flats, etc. The large environment shall represent a huge community, e.g., an online recommender, an online shop, etc. The usage of the two different sizes of the dataset shall show that the proposed system is not limited by the size of the community.

The objectives of this thesis are:

- Prediction accuracy improvement
- Research and development of a new recommendation system
- Evaluation by the usage of small and large datasets

1.2.1 Prediction accuracy improvement

The first objective of this thesis is the improvement of the predictions' accuracy. The system shall be able to predict entries within the user-item matrix as exactly as possible. Since the predictions are calculated by the usage of the collaborative-filtering algorithms which deliver the similarity between users or items, this thesis presents state-of-the art and newly developed collaborative-filtering algorithms which calculate these similarities. Therefore the presented thesis focuses on the improvement of the similarity calculation to improve the predictions' accuracy.

1.2.2 Research and development of a new recommendation system

The quality of the calculated predictions which consider the similarities of the collaborative-filtering algorithms can be exploited by the usage of error rates. However, the main task of this thesis is the researching and development of a new recommendation system which is able to reduce the error rates, such as the MAE, MSE, and RMSE. The new recommendation system shall be able to reduce the error rates compared to existing recommendation systems which use collaborative-filtering techniques.

1.2.3 Evaluation by the usage of small and large datasets

The proposed system shall be able to consider small and large datasets. A small dataset represents e.g., a family, a block of flats, etc. The large dataset represents the preferences of e.g, an online recommender, an online shop, etc. The evaluation of the presented thesis considers a small dataset which is the result of an undertaken survey and a large dataset from MovieLens. The proposed system shall be able to consider both datasets and the evaluation shall prove that the system is not limited to the size of the dataset.

1.3 Contribution

The contribution tackles the objectives of this thesis which includes the improvement of the predictions' accuracy, the research and development of a new recommendation system, and the evaluation of this new system. Basically this thesis will present a dynamic multi-algorithm collaborative-filtering system which finds the most accurate algorithm dynamically. The single parts of the contribution are described below.

This system uses SotA filtering algorithms. These SotA collaborative-filtering algorithms are described in Section 2.2.2.1 in more detail. In addition the dynamic multi-algorithm collaborative-filtering system also includes newly developed algorithms, which overcome researched weaknesses of the mentioned SotA existing algorithms. Section 4.3.1 presents these newly developed algorithms. The evaluation of this thesis proves that the newly developed collaborative-filtering algorithms which overcome researched weaknesses are able to deliver more accurate similarities. Since the prediction calculation uses these similarities, the predictions' accuracy is improved.

The main contribution of this thesis is the dynamic selection of the most accurate collaborative-filtering algorithm, which is described in Section 4.3. The system takes a selection of multiple algorithms into account and selects the most accurate one for the currently active user or item. This proposed algorithm is identified by using the following approach. At the beginning, the proposed dynamic multi-algorithm collaborative-filtering system selects the active user or item. This active user or item is used to find the k-nearest neighbours. The k-nearest neighbours includes the users or items which are quite similar to the active user or item. This neighbourhood is used to create a user-item matrix which contains the ratings from users on items. The user-item matrix is used to calculate predictions of the entries within the matrix. These predictions are used to exploit error rates, such as the MAE, the MSE, and the RMSE. The entire procedure is accomplished by taking each collaborative-filtering algorithm into account which is included into the proposed system. The output of the dynamic multi-algorithm collaborative-filtering system is the most accurate filtering algorithm, which is strongly connected to the active user or item and its neighbourhood.

However, the evaluation of this thesis proves that the proposed system is able to reduce the error rates significantly compared to existing recommendation systems from other researchers. The evaluation considers a small dataset. This small dataset has been built by the usage of the results from a survey which has been accomplished at the THM. Besides the small dataset, the experiments also consider a large dataset from MovieLens. The results prove that the proposed dynamic multi-algorithm collaborative-filtering system is not limited to small datasets. It can also be used for larger datasets.

1.4 Organization of the Thesis

This thesis is organized as follows: Chapter 2 introduces the reader to existing recommender systems. It is split into two main parts. The first part describes the user profiling. The second part describes the collaborative filtering approach. In addition, hybrid systems, which includes content-based and collaborative filtering techniques, are presented too. Since the main task of this thesis focuses on collaborative filtering algorithms, the content-based filtering techniques and the hybrid systems are described briefly. Chapter 3 presents the used methodology for the achieving of the results in this thesis. Chapter 4 introduces the reader to the developed and researched recommendation system. This section is divided into three main parts. The first part tackles the creation of the user profiles, which is described in Section 4.1. The second part, which is presented in Section 4.2, describes the content-based algorithms and the third part introduces the reader to the proposed dynamic multi-algorithm collaborative-filtering system, which is presented in Section 4.3. The evaluation of the proposed dynamic multi-algorithm collaborative-filtering algorithm is presented in Chapter 5. The evaluation considers the dataset from the survey and the dataset from MovieLens. Chapter 6 briefly introduces the reader into the developed Personal Program Guide (PPG), which is responsible for the visualization of the recommendations and the setting of the preferences. Finally, Chapter 7 concludes the thesis and presents possible future work.

Chapter 2

Background and Related Work

With the introducing of the Internet in the 1990s users were able to get an immense variation of information. The result of the immense number of information is that users are overloaded with information [3]. During this time recommendation systems became important. Recommendation systems are used to filter the available information and present recommendations which fit to users' preferences. These kind of systems are used in several fields. They are used by online shops, by large-scale image libraries, or by movie databases.

An example for a large-scale image library is Flickr which is a web page that shares images. Due to the immense number of images a searching for images with a special topic like landscape could be difficult. The paper from Jianping Fan *et al.* [41] tackles this problem. The system from the authors automatically generates a topic network which summarizes the large-scale collections of the images from Flickr at a semantic level. Additionally the system uses a hyperbolic visualization which enables an interactive navigation and exploration of the topic network e.g., users are able to select a image topic. The queries are used to search images within the topic network and to recommend most representive images for the given image topic.

Online shops also offers a lot of products. Amazon is an example for a huge online shop. They offer books, CDs, DVDs, clothes, TVs, and many more. In order to support users to find goods that could be interesting to them, Amazon generates recommendations. The industry paper from Greg Linden *et al.* [42] presents the technique which is used to generate the recommendations on the Amazon web page. Amazon uses collaborative-filtering techniques for the creation of the recommendations. The recommendations are strongly connected to the interest of an individual user. Amazon matches each item of a user to similar items and combines those items into a recommendation list. The finding of similar items is realized by the usage of the *Cosine Similarity* which is described below.

However, this thesis tackles the creation of recommendations by the usage of video content, such as DVB content and movies from MovieLens. The following sections present the related work which tackles the topics of this thesis. They are split into three main parts.

- User profiling
- Filtering
  - Content-based filtering
  - Collaborative filtering
  - Hybrid recommendation systems

- Presentation of recommendations

The main part of this thesis is the research and the evaluation of a newly developed dynamic multi-algorithm collaborative-filtering system. However, since the proposed system needs user profiles for the creation of recommendations, this chapter will also briefly introduce the reader to this topic. In addition, the presented recommendations are based on the content-based filtering and the collaborative-filtering approaches. Therefore, this chapter also briefly presents related work in this field. Finally, the recommendations are presented within an interface. This chapter introduces some existing interfaces which are able to present recommendations.

2.1 User Profiling

User profiles contain data from a user. A user profile can contain various data about a user, e.g. name of the user, set preferences, education, demographic information, and so forth. These profiles can be used to generate individual recommendations which are based on these profiles. The creation of the user profiles can be created in an implicit or explicit manner [43, 44]. The presented thesis takes these approaches into account. The following publications present related works in this field.

The recommendation system from De Pessemier *et al.* [45] uses metadata for the recommendation creation. The metadata can contain a genre, a director, a keyword, a title, an actor, a coworker, the spoken language, or the caption language. The authors make use of so-called metadata terms t_i. Examples of these terms are “soccer”, “Antonio Banderas”, “violence”, etc. Each of these terms belongs to a field $f_i \in \{$*Genre, Actor, Director, Coworker, Keyword, Spoken Language, Title, Caption*$\}$. The system associates each of these terms with a user appreciation that is defined as u_i. This user appreciation is in the range [-1,1]. The user profiles are saved in the database in a form of 3-tuples defined as (t_i, f_i, u_i). The system ranks the importance of the used metadata. For example, a genre is more significant than a keyword. In order to take this importance into account, the authors assign an importance factor W_i for each field f_i. The system updates the 3-tuples in an explicit and implicit manner. The implicitly created user profile is created by logging the viewing behaviour. The system logs the time the user spends watching a video. The explicit user profile is updated by setting ratings. In addition, the user appreciation is updated by using Equation 2.1 if a 3-tuples is already in the user profile. Otherwise Equation 2.2 is used to create the user appreciation [45].

$$\acute{u} = (1 - \alpha) \cdot u + \alpha \cdot \beta \tag{2.1}$$

Where $\acute{u}$ represents the new user appreciation of t_i, u stands for the old user appreciation of the term and α is a parameter which specifies the learning rate and is in a range between 0 and 1. β is in a range [-1,1] and represents the score from the implicit and explicit rating mechanism.

$$\acute{u} = \beta \tag{2.2}$$

The system from De Pessemier *et al.* extracts the information from the TV-Anytime metadata [46]. After this step the system checks which terms t_i are available in the user profile and the system calculates a recommendation score that is defined by Equation 2.3 [45].

$$S = \frac{\sum_i u_i \cdot W(f_i)}{\sum_i W(f_i)} \tag{2.3}$$

Figure 2.1 illustrates the procedure of the recommendation score calculation in more detail.

In contrast to the paper from De Pessemier *et al.*, this thesis only uses specified metadata, which is sent within the DVB Transport Stream, for the creation of the user profiles. The usage of the metadata which is sent within the DVB Transport Stream guarantees that only specified genres are used for the creation of the user profiles. This behaviour helps to create the recommendations by considering the collaborative-filtering techniques because each user profile is built by taking the same pool of genres into account.

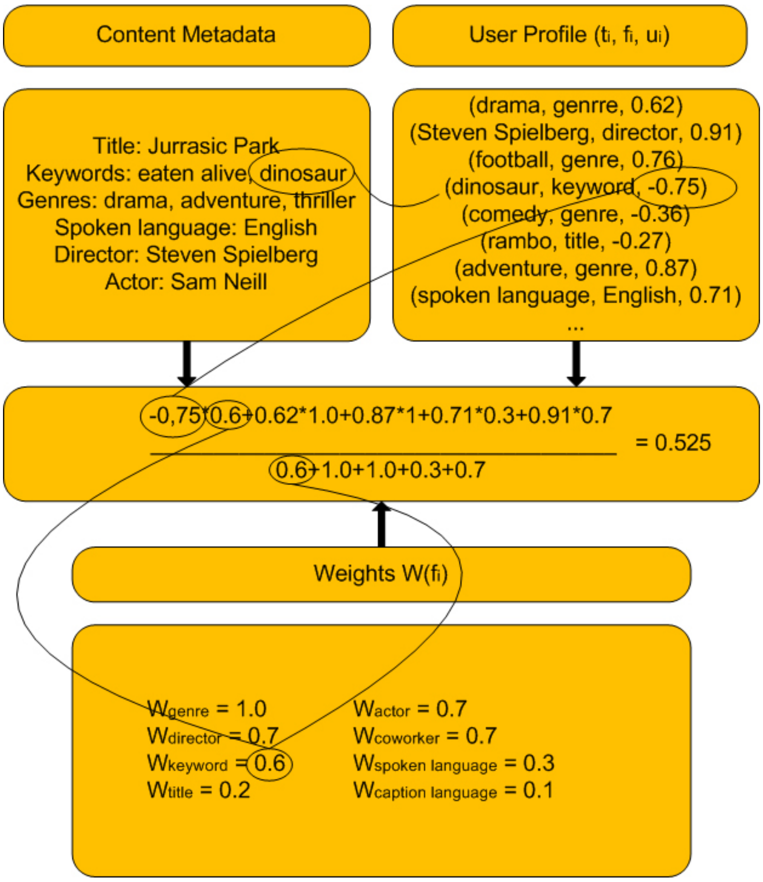


Figure 2.1: Procedure of the recommendation score calculation defined by De Pessemier *et al.* [45]

The system from Hopfgartner *et al.* [47] takes the inverse exponential weighting from Campbell *et al.* [48] into account which is presented by Equation 2.4.

$$a_j = \frac{1 - C^{-j+1}}{\sum_{k=2}^{j_{max}} 1 - C^{-k+1}} \tag{2.4}$$

Hopfgartner *et al.* uses this approach within their system. They define C as a category, j defines the iterations and k is the number of the clusters. With this approach events, like a movie, a documentation, or a soap, that are added recently get a higher weighting. The created recommendations are based on current interests. Therefore the recommendations represent the current preferences of the users. This behaviour could create a problem. For example, a user likes to watch soccer, which is broadcast every Saturday. After six months the soccer season takes a break for three months. After three months, the proposed system from Hopfgartner *et al.* will rate soccer quite low, because the user has not watched it for several month. In contrast to this paper, the presented approach of the thesis will decrease the implicitly logged Recommendation Index (RI), but the explicit settings will also be used for the creation of the recommendations.

Zhang and Zheng [49] propose a system which is based on TV-Anytime metadata. TV-Anytime metadata can contain information like title, genre, synopsis, actors, directors, etc. This kind of searching uses the content-based approach. The system is able to search events by using the metadata that are saved in the TV-Anytime format. The authors also introduces two calculations, the *Average Content Affinity* (ACA) and the *Category Affinity Ratio* (CAR). The ACA determines the average affinity for a special category. The CAR represents the affinity of a user on a particular instance that is related to other instances of the currently used category. Both calculations, the ACA and the CAR, are based on the usage history. Therefore the user profiles are implicitly created.

2.2 Filtering

A recommendation system needs some kind of filtering techniques for the creation of recommendations. Basically three kinds of filtering approaches exist - the content-based filtering, collaborative filtering and hybrid approaches [50]. These three different approaches are described in the following sections. Since the main topic of this thesis tackles the collaborative-filtering approach, the content-based and the hybrid approach will be described only briefly.

2.2.1 Content-Based Filtering

Recommendation systems, which are based on content-based filtering techniques, use the metadata from the content for the creation of recommendations [51, 52]. For example, users can set their preferences in an explicit manner. They can set that they prefer a specific movie, like “It” from Stephen King. The recommendation system can use this information for the creation of the recommendations. The system can search for this title or for this director and recommend movies from Stephen King.

2.2.2 Collaborative Filtering

In contrast to content-based filtering, collaborative-filtering techniques use the data from a community for the creation of recommendations [52–55]. The system searches similar users or items [56–60], e.g. movies, and creates the recommendations based on these similarities.

The collaborative filtering can be divided into two different approaches, the model-based and the memory-based approach [61].

The model-based approach considers just parts of the information. This approach develops a model of user ratings. The model building is realized by machine learning algorithms, such as the Bayesian network, the clustering approach, or other rule-based approaches [7]. The Bayesian network formulates a model for probabilistic collaborative filtering [62]. Clustering models clusters data into similar items/users [62–64].

The memory-based approach uses the entire information from the user-item matrix, which contains the ratings from all users on selected items, e.g. movies. Systems, which use this approach typically try to find similar users or items - the so-called neighbourhood [7, 61, 65]. This technique is widely used and more popular [10].

Besides these techniques most recommendation systems from other researchers distinguish two classes of collaborative-filtering algorithms, the user-based and the item-based [66]. The following sections will describe SotA algorithms which can be used to calculate similarities.

2.2.2.1 Collaborative-Filtering Algorithms

This section presents existing collaborative-filtering algorithms, such as the *Pearson-r Correlation*, the *Spearman Rank Correlation*, the *Cosine Similarity*, and the *Adjusted Cosine Similarity*. Besides the presentation of these algorithms, this section presents weaknesses of them.

Additionally this section presents the Weighted Sum approach which is used to calculate the predictions. The k-nearest neighbour approach and the calculation of error rates is described as well.

2.2.2.1.1 Pearson-r Correlation

The Pearson-r Correlation calculates the linear correlation between two objects [7, 9–11, 31, 36, 67]. It takes only the co-rated items into account. The co-rated items are the items that were rated by two users. Table 2.1 illustrates some item ratings.

	User 1	User 2
Item 1	5	5
Item 2	4	
Item 3	3	3
Item 4	5	0
Item 5	0	4
Item 6		0
Item 7	1	2
Item 8	3	4
Item 9	2	4
Item 10	2	2
Item 11	1	
Item 12	1	3

Table 2.1: Ratings from two users on twelve items where some of the items are not rated by both users

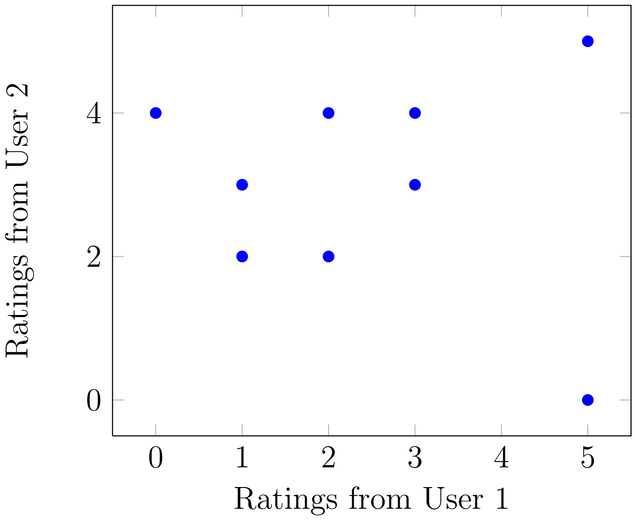


Figure 2.2: The Pearson-r Correlation between User 1 and User 2 by considering the item ratings

Since Item 2, Item 6, and Item 11 are not rated by two users, the *Pearson-r Correlation* will not take these items into account. The results of the *Pearson-r Correlation* can be in the range [-1;+1]. -1 represents full negative linear correlation, +1 full positive linear correlation, and 0 no linear correlation between the considered objects. Figure 2.2 illustrates the linear correlation between User 1 and User 2 and Figure 2.3 shows possibilities of other linear correlations graphically.

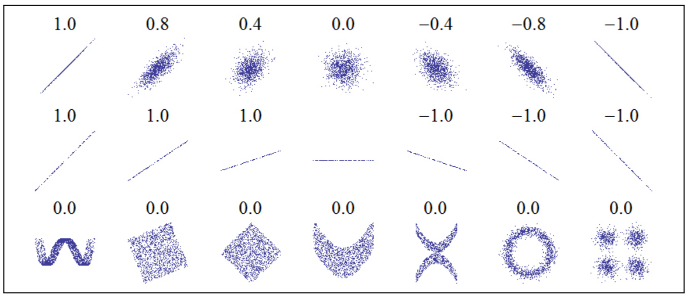


Figure 2.3: Graphical overview of the linear correlations which can be achieved by the usage of the Pearson-r Correlation [68]

Equation 2.5 defines the *Pearson-r Correlation* if the linear correlation between two items shall be calculated. Equation 2.6 defines the *Pearson-r Correlation*, if the user-based approach shall be used.

$$PC_{sim}(i,j) = \frac{\sum_{\mathrm{u}\in\mathrm{U}}(R_{u,i}-\overline{R}_i)(R_{u,j}-\overline{R}_j)}{\sqrt{\sum_{\mathrm{u}\in\mathrm{U}}(R_{u,i}-\overline{R}_i)^2}\sqrt{\sum_{\mathrm{u}\in\mathrm{U}}(R_{u,j}-\overline{R}_j)^2}} \tag{2.5}$$

$PC_{sim}(i,j)$ is the linear correlation between item i and item j. $R_{u,i}$ is the rating from user u of item i. $R_{u,j}$ is the rating from user u of item j. $\overline{R}_i$ is the average of the ratings from item i and $\overline{R}_j$ is the average of the ratings from item j. $u \in U$ is the summation of the users who rated both items i and j.

$$PC_{sim}(u,v) = \frac{\sum_{\mathrm{i}\in\mathrm{I}}(R_{u,i}-\overline{R}_u)(R_{v,i}-\overline{R}_v)}{\sqrt{\sum_{\mathrm{i}\in\mathrm{I}}(R_{u,i}-\overline{R}_u)^2}\sqrt{\sum_{\mathrm{i}\in\mathrm{I}}(R_{v,i}-\overline{R}_v)^2}} \tag{2.6}$$

$PC_{sim}(u,v)$ is the linear correlation between user u and user v. $R_{u,i}$ is the rating from user u of item i. $R_{v,i}$ is the rating from user v of item i. $\overline{R}_u$ is the average of the ratings from user u and $\overline{R}_v$ is the average of the ratings from user v. $i \in I$ is the summation of the items that are rated by the users u and v.

Weakness

The main problem of the *Pearson-r Correlation* is the calculation of the linear correlation. For example, if one user rates five items with the values $u1 = \{1, 2, 3, 4, 5\}$ and a second user rates the five items with the values $u2 = \{10, 11, 12, 13, 14\}$, the linear correlation between these two users is 1, although the ratings are quite different. The equations in the Section 4.3.1.1 overcome this weakness.

2.2.2.1.2 Spearman Rank Correlation

The *Spearman Rank Correlation* is based on the *Pearson-r Correlation* [7, 32, 69]. In contrast to the *Pearson-r Correlation*, the *Spearman Rank Correlation* converts the entries within the user-item matrix into ranks. The entire procedure of the rank building is shown by Table 2.2. This procedure considers the following ratings from a user *user={5,4,3,5,0,2,1,3,2,2,1,1}*.

In this example the first value from the ratings gets the highest index. At the beginning the procedure sorts the indexes by values in a decreasing order. This sorting is shown by the columns with the "Sort by value" header. After this step the system ranks the values as shown by the columns with the "Get rank" header. The last step of the procedure sorts the values and ranks by indexes.

Input		Sort by value		Get rank				Sort by index		
Index	Value	Index	Value	Index	Value	Rank	Normalized	Index	Value	Rank
12	5	12	5	12	5	12		12	5	11.5
11	4	9	5	9	5	11	(12+11)/2=11.5	11	4	10
10	3	11	4	11	4	10	10/1=10	10	3	8.5
9	5	10	3	10	3	9		9	5	11.5
8	0	5	3	5	3	8	(9+8)/2=8.5	8	0	1
7	2	7	2	7	2	7		7	2	6
6	1	4	2	4	2	6		6	1	3
5	3	3	2	3	2	5	(7+6+5)/3=6	5	3	8.5
4	2	6	1	6	1	4		4	2	6
3	2	2	1	2	1	3		3	2	6
2	1	1	1	1	1	2	(4+3+2)/3=3	2	1	3
1	1	8	0	8	1	1	1/1=1	1	1	3

Table 2.2: The procedure of the rank building by the usage of the Spearman Rank approach

Table 2.3 shows the ratings from two users and the ranks that are built by using the *Spearman Rank Correlation.* These ranks are used to calculate the similarities between two objects (e.g. users or items) by the usage of the *Pearson-r Correlation.* The results are in the range [-1;+1], where +1 is a positive, -1 a negative relationship, and 0 no relationship between two objects. Equation 2.7 defines the Spearman Rank Correlation if the item-based approach is used and Equation 2.8 is used if the user-based approach is used.

	User 1	Ranks User 1	User 2	Ranks User 2
Item 1	5	11.5	5	12
Item 2	4	10	4	9.5
Item 3	3	8.5	3	6
Item 4	5	11.5	0	1.5
Item 5	0	1	4	9.5
Item 6	2	6	0	1.5
Item 7	1	3	2	3.5
Item 8	3	8.5	4	9.5
Item 9	2	6	4	9.5
Item 10	2	6	2	3.5
Item 11	1	3	3	6
Item 12	1	3	3	6

Table 2.3: The ratings from two users on twelve items and the ranks which are created by the usage of the Spearman Rank Correlation

$$SRC_{sim}(i,j) = \frac{\sum_{\mathrm{u}\in\mathrm{U}}(Rg_{u,i}-\overline{Rg}_i)(Rg_{u,j}-\overline{Rg}_j)}{\sqrt{\sum_{\mathrm{u}\in\mathrm{U}}(Rg_{u,i}-\overline{Rg}_i)^2}\sqrt{\sum_{\mathrm{u}\in\mathrm{U}}(Rg_{u,j}-\overline{Rg}_j)^2}} \tag{2.7}$$

$SRC_{sim}(i,j)$ is the rank correlation between item i and item j. $Rg_{u,i}$ is the rank-rating from user u of item i. $Rg_{u,j}$ is the rank-rating from user u of item j. $\overline{Rg}_i$ is the average of the rank-ratings from item i and $\overline{Rg}_j$ is the average of the rank-ratings from item j. $u \in U$ is the summation of the users who rated both items i and j.

$$SRC_{sim}(u,v) = \frac{\sum_{\mathrm{i \in I}}(Rg_{u,i} - \overline{Rg}_u)(Rg_{v,i} - \overline{Rg}_v)}{\sqrt{\sum_{\mathrm{i \in I}}(Rg_{u,i} - \overline{Rg}_u)^2}\sqrt{\sum_{\mathrm{i \in I}}(Rg_{v,i} - \overline{Rg}_v)^2}} \tag{2.8}$$

$SRC_{sim}(u,v)$ is the rank correlation between user u and user v. $Rg_{u,i}$ is the rank-rating from user u of item i. $Rg_{v,i}$ is the rank-rating from user v of item i. $\overline{Rg}_u$ is the average of the rank-ratings from user u and $\overline{Rg}_v$ is the average of the rank-ratings from user v. $i \in I$ is the summation of the items that are rated by the users u and v.

In contrast to the *Pearson-r Correlation*, the *Spearman Rank Correlation* does not calculate the linear correlation between two objects, like users or items. It assesses whether the relationship can be described as a monotonic function, which is shown in Figure 2.4.

The figure shows a perfect monotonic function, since each of the variables is a perfect monotonic function of each other. In this case, the *Spearman Rank Correlation* is +1 or -1. The *Pearson-r Correlation* would not be +1 or -1, because it is not a perfect linear correlation.

Weakness

The *Spearman Rank Correlation* is based on the *Pearson-r Correlation*, which calculates the linear correlation between two objects as described in Section 2.2.2.1.1. The weakness of the *Pearson-r Correlation* is described in Section

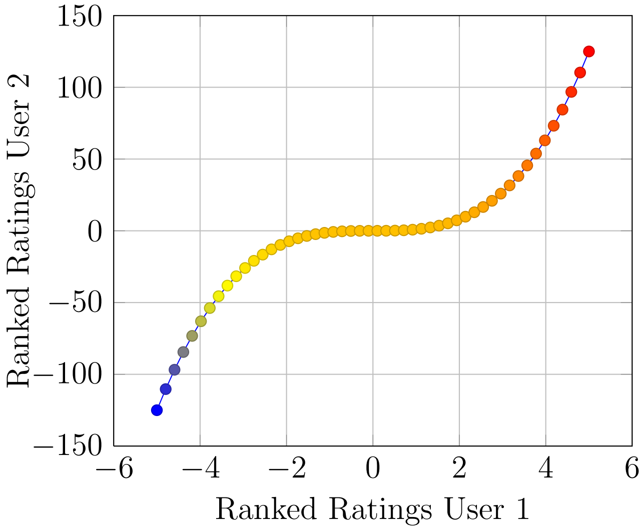


Figure 2.4: Spearman Rank Correlation - monotonic function between the ranked ratings two users

2.2.2.1.1. The equations in Sections 4.3.1.2 and 4.3.1.3 analyse this problem and try to overcome it.

2.2.2.1.3 Cosine Similarity

The cosine angle between two vectors can be computed by the usage of the *Cosine Similarity* [10, 31, 36, 38]. Figure 2.5 presents two vectors in a two-dimensional room. α is the cosine angle between these two vectors. The results of the *Cosine Similarity* are in the range [0;1], where 1 represents full similarity, and 0 no similarity between the angle of the two vectors. The calculation which takes the item-based approach into account is defined by Equation 2.9. Equation 2.10 is used if the user-based approach is considered.

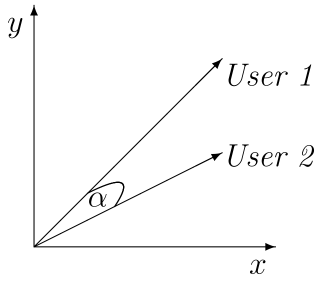


Figure 2.5: Two vectors which are created by the usage of the user's ratings in a two-dimensional room

$$CS_{sim}(i,j) = \frac{\vec{i} \cdot \vec{j}}{\left\|\vec{i}\right\| \cdot \left\|\vec{j}\right\|} \tag{2.9}$$

$CS_{sim}(i,j)$ is the similarity between the two vectors i and j. $\vec{i}$ represents the vector of i and $\vec{j}$ is the vector of j. $\vec{i} \cdot \vec{j}$ is the dot product from vector i and vector j. $\left\|\vec{i}\right\|$ is the magnitude of vector i and $\left\|\vec{j}\right\|$ is the magnitude of vector $\vec{j}$.

$$CS_{sim}(u,v) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \cdot \|\vec{v}\|} \tag{2.10}$$

$CS_{sim}(u,v)$ is the similarity between the two vectors u and v. $\vec{u}$ represents the vector of u and $\vec{v}$ is the vector of v. $\vec{u} \cdot \vec{v}$ is the dot product from vector u and vector v. $\|\vec{u}\|$ is the magnitude of vector u and $\|\vec{v}\|$ is the magnitude of the vector $\vec{v}$.

Weakness

The *Cosine Similarity* computes the cosine angle between two vectors, but it does not consider the length of the vectors. For example, User 1 ranks five items $u1 = \{1,1,1,1,1\}$ and User 2 ranks five items $u2 = \{5,5,5,5,5\}$. The cosine similarity between these two users will be 1, although they ranked the items quite differently. The equations presented in Sections 4.3.1.7, 4.3.1.8, 4.3.1.9, 4.3.1.10, 4.3.1.11, and 4.3.1.12 tackle this problem and try to overcome it.

2.2.2.1.4 Adjusted Cosine Similarity

In contrast to the above mentioned approaches, the *Adjusted Cosine Similarity* considers the average of the ratings from users/items [10,12]. The results of the *Adjusted Cosine Similarity* are in the range [-1;+1]. -1 represents a full negative similarity, 0 no similarity, and 1 full positive similarity. Equation 2.11 is used if the item-based approach is used. Equation 2.12 performs the calculation if similarities between users shall be calculated.

$$AJCS_{sim}(i,j) = \frac{\sum_{\mathrm{u} \in \mathrm{U}} (R_{u,i} - \overline{R}_u)(R_{u,j} - \overline{R}_u)}{\sqrt{\sum_{\mathrm{u} \in \mathrm{U}} (R_{u,i} - \overline{R}_u)^2} \sqrt{\sum_{\mathrm{u} \in \mathrm{U}} (R_{u,j} - \overline{R}_u)^2}} \tag{2.11}$$

$AJCS_{sim}(i,j)$ is the correlation between item i and item j. $R_{u,i}$ is the rating from user u of item i. $R_{u,j}$ is the rating from user u of item j. $\overline{R}_u$ is the average of the ratings from user u. $u \in U$ is the summation of the users who rated both items i and j.

$$AJCS_{sim}(u,v) = \frac{\sum_{\mathrm{i} \in \mathrm{I}} (R_{u,i} - \overline{R}_i)(R_{v,i} - \overline{R}_i)}{\sqrt{\sum_{\mathrm{i} \in \mathrm{I}} (R_{u,i} - \overline{R}_i)^2} \sqrt{\sum_{\mathrm{i} \in \mathrm{I}} (R_{v,i} - \overline{R}_i)^2}} \tag{2.12}$$

$AJCS_{sim}(u,v)$ is the correlation between user u and user v. $R_{u,i}$ is the rating from user u of item i. $R_{v,i}$ is the rating from user v of item i. $\overline{R}_i$ is the average of the ratings from item i. $i \in I$ is the summation of the items that are rated by the users u and v.

2.2.2.2 Existing Systems and Approaches

This section briefly presents the related studies of existing systems and approaches, which use collaborative-filtering algorithms.

Herlocker *et al.* [7] apply an analysis framework that divides the neighbourhood-based prediction approach into three components. The three components are a similarity computation, a neighbour selection, and a rating combination. A neighbour selection, also known as the k-nearest neighbour approach, is described below. This approach is used from several researchers. Table 2.4 presents possible similarity results, which are achieved by computation the similarities between the active user/item and all other users/items from a given user-item matrix. The table contains the number/position of the user within the user-item matrix and the similarity result.

User/Item Number	**Similarity**
1	0.65
2	0.77
3	-0.12
4	0.98
5	0.31
6	0.49
7	0.85
8	-0.79
9	0.16
10	-0.37
11	0.73
12	-0.02

Table 2.4: The similarity values between an active user or item and other users/items

Within the next step, a system can order the achieved similarities in decreasing order. Table 2.5 presents the five-nearest neighbours. The authors use data from the movie recommendation site MovieLens. The calculations of similarities are realized by the *Pearson-r Correlation*, *Spearman Rank Correlation*, and the *Mean Squared Difference*. The evaluation of the system uses the MAE which is defined by Equation 2.13.

User/Item Number	**Similarity**
4	0.98
7	0.85
2	0.77
11	0.73
1	0.65

Table 2.5: The similarity values between an active user or item and other users/items which are ranked in decreasing order

$$MAE = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N} \tag{2.13}$$

p_i is the prediction and q_i is the true value. Their experiments show that the performance of the *Pearson-r Correlation* and the *Spearman Rank Correlation* are better than those of the *Mean Squared Difference*. The *Pearson-r Correlation* and the *Spearman Rank Correlation* have almost the same results (MAE of ≈ 0.74), but the authors recommend the *Pearson-r Correlation* for performing the calculations. The authors do not consider the user-based approach. In addition, the authors do not present the results of the tests by using the *Cosine Similarity*. The thesis presents the results, which have been achieved by using the other collaborative-filtering techniques as well.

Sawar *et al.* [10] consider the *Pearson-r Correlation*, the *Cosine Simi-*

larity, and the *Adjusted Cosine Similarity* for the calculation of similarities. The authors use the item-based approach for their calculations. The evaluation is realized by the usage of the dataset from MovieLens [70]. This dataset is used to create reduced user-item matrices randomly. The accuracy of the predictions is exploited by the usage of the MAE [10, 11, 13]. The authors use the *Weighted Sum* [9–11] approach for the prediction calculation. This approach is defined by Equation 2.14.

$$P_{a,i} = \overline{r}_a + \frac{\sum_{\mathrm{u} \in \mathrm{U}} (r_{u,i} - \overline{r}_u) \cdot sim_{a,u}}{\sum_{\mathrm{u} \in \mathrm{U}} |sim_{a,u}|} \tag{2.14}$$

$P_{a,i}$ is the prediction of the active user a for item i (e.g. a genre or a movie). $\overline{r}_a$ represents the average of the ratings of the active user a. $r_{u,i}$ is the rating of user u for item i. $\overline{r}_u$ represents the average of the ratings from user u without the rating of item i. $sim_{a,u}$ represents the similarity between the active user a and user u. This approach can also be used for the item-based approach. It is equivalent. The evaluation of this paper shows that the *Adjusted Cosine Similarity* produces the lowest MAE. It delivers a MAE of ≈ 0.72. In contrast to this paper, the presented thesis also considers the user-based approach. In addition, the thesis also presents newly developed algorithms that are able to produce a lower error rate than the system from these authors.

The paper from Papegelis and Plexousakis [11] uses the *Pearson-r Correlation* by considering the user-based and the item-based approach. The authors take user profiles into account that are created by explicit settings and implicitly logged viewing behaviour. The paper compares the algorithms by using different levels of sparsity and different thresholds. The evaluation of this paper shows that the predictions that have been achieved by taking the item-based approach into account, which have been derived by explicitly set preferences, deliver the best performance. The system of Papegelis and Plexousakis delivers a MAE of ≈ 0.84. However, this paper does not compare the results with other widely used algorithms. It only considers the *Pearson-r Correlation*. The presented thesis considers a selection of multiple algorithms, evaluates and compares them. In addition the proposed system delivers an error rate, which is below the system from the mentioned authors.

Krishnan *et al.* [15] compare results from an online study of humans with the online recommender system from MovieLens. Basically the paper presents a comparison between a "personal" recommender system and a "impersonal" recommender system. The authors evaluate their results by the calculation of the MAE. The system produces a MAE of ≈ 0.87. In addition, the results of their evaluation shows that MovieLens scores the overall MAE better than the personal preferences. In some cases the "personal" system produces better results. The authors use the *Cosine Similarity* for the calculation of the similarities by taking the item-based approach into account. However, in contrast to this paper, the presented thesis uses several collaborative-filtering algorithms. In addition, the thesis also presents the results that are achieved by the calculation of the MSE and the RMSE. The MSE is defined by Equation 2.15 and the RMSE is defined by Equation 2.16.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(p_i - q_i)^2 \tag{2.15}$$

p_i is the prediction and q_i is the true value.

The RMSE includes the MSE for its calculation. It is the square root of the MSE.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(p_i - q_i)^2} \tag{2.16}$$

p_i is the prediction and q_i is the true value.

The paper from Martín-Vincente *et al.* [34] uses a semantic approach, which builds implicit trust networks that can be applied in collaborative recommendation systems. It obtains trust relations from a record of results by considering previous recommendations and by exploiting the interaction with the system. The authors use a TV ontology that is presented in a paper from Blanco-Fernandez *et al.* [71]. The proposed system from the authors uses the information to build relations between the users. Unfortunately the authors do not compare their results with other existing systems.

The paper from Zhang *et al.* [72] presents a regression procedure that uses the matrix factorization. The usefulness of their system is proved by using the datasets from MovieLens and Yahoo. The system from the authors is able to reduce the RMSE to 0.8777 by considering the dataset from MovieLens. In contrast to this paper, the presented thesis focuses on the improvement of the predictions' accuracy by using the similarity calculation between users or items. The thesis does not use the matrix factorization. However, the proposed system of this thesis is able to produce a RMSE which is significantly below the error rate of the system from these authors.

The recommendation system from Liang-hao and Lin-hao [40] is based on users. The system uses the *Cosine Similarity* and the *Pearson-r Correlation* for the similarity calculation. The similarities are used to calculate predictions by using the *Weighted Sum* approach. The authors also use the k-nearest neighbour approach, which finds the neighbourhood that contains similar users. The exploiting of the system is realized by the usage of the MAE. The authors evaluate their system with a dataset from MovieLens. The system of the authors produces a MAE of ≈ 0.80. However, in contrast to this paper, the proposed system of this thesis uses a dynamic selection of the most accurate collaborative-filtering algorithm. Besides this fact the thesis also presents results by using the item-based approach. In addition the evaluation of the thesis proves that the error rate of the proposed system is significantly lower than the error rate from the system of Liang-hao and Lin-hao.

The system from Cao *et al.* [14] uses a novel matrix factorization system. This system includes an efficient learning algorithm and prediction strategies. The authors use datasets from MovieLens, Netflix, and EachMovie to evaluate their system. The evaluation is based on the calculation of the MAE. The system delivers a MAE of ≈ 0.75. However, the proposed system of this thesis is able to deliver a lower MAE compared to the system from Cao *et al.*

In terms of collaborative-filtering systems the Netflix competition was a big event. Netflix initiated a competition in 2006. The aim of this competition was the improvement of the predictions' accuracy of collaborative filtering systems. The challengers were asked to decrease the RMSE of the calculated predictions. The competition ended in 2009. Netflix offers a process prize of 50,000 USD every year. The grand prize in 2009 was 1,000,000 USD. The winner of this competition was able to reduce the RMSE to 0.8567. The following publications consider the dataset from Netflix and present their approaches.

The system from Bell and Koren [18] enhances the neighbour-based approach. It removes so-called "global effects". This approach helps the authors to make the ratings more comparable. The result of this approach is the improvement of the predictions' accuracy. The authors also present a simultaneous derivation of interpolation weights. The k-nearest neighbours are identified by using the *Pearson-r Correlation*, which calculates similarities between objects. The system from the authors is able to reduce the RMSE to 0.8982.

The system from Töscher *et al.* [73] combines the regularized matrix factorization with the k-nearest neighbour approach. The similarities between the used data from Netflix is calculated by a variation of the *Pearson-r Correlation*. The proposed system from the authors is able to produce a RMSE of 0.9042.

Wen [17] uses the *Adjusted Cosine Similarity* for the similarity calculation by considering the dataset from MovieLens. The similarities are used to find the k-nearest neighbours. Besides the *Adjusted Cosine Similarity*, the system also takes an *Item-Based EM* and the *Sparse Singular Value Decomposition*

(SVD) into account. In addition, the system from Wen performs some so-called postprocessing tricks, which are able to decrease the RMSE as well. The system is able to reduce the RMSE to 0.8930 by using a blending of *Item-Based EM* and the *Sparse SVD*. The author only considers the item-based approach. In contrast to this paper, the presented evaluation also considers the results by using the user-based approach.

The paper from Töscher *et al.* [16] presents the results of the system that won the grand prize of the Netflix competition. The paper compares different approaches, which have been researched by the authors. The authors presents the results by using the k-nearest neighbour approach in combination with collaborative-filtering algorithms, such as the *Pearson-r Correlation*, the *Spearman Rank Correlation*, the *Set Correlation*, the *MSE Correlation*, and the *Ratio Correlation*. In addition to these techniques, the authors also researched other kinds of techniques, like several kinds of SVD and matrix factorizations. The lowest RMSE was achieved by using the matrix factorization. The authors were able to reduce the RMSE to 0.8567.

The main difference of other research works and the proposed system in this thesis is the dynamic choice of the most accurate collaborative-filtering algorithm. The system is able to select the most accurate filtering algorithm, which is strongly connected to the active user/item and its neighbourhood. To the best of my knowledge, the proposed approach has never been realized before.

Table 2.6 presents some error rates from the related work. The results of the presented evaluation prove that the proposed dynamic multi-algorithm collaborative-filtering system of this thesis is able to deliver an error rate

Author	**Algorithm**	**MAE**	**RMSE**	**Dataset**
Sawar *et al.* [10]	Adjusted Cosine and neighbour selection	≈ 0.72		MovieLens
Papegelis and Plexousakis [11]	Pearson-r	≈ 0.84		own dataset
Herlocker *et al.* [7]	Pearson-r and neighbour selection	≈ 0.74		MovieLens
Krishnan *et al.* [15]	Cosine	≈ 0.87		MovieLens
Liang-hao and Lin-hao [40]	Cosine and Pearson-r and neighbour selection	≈ 0.80		MovieLens
Cao *et al.* [14]	Matrix Factorization	≈ 0.75		MovieLens
Zhang *et al.* [72]	Matrix Factorization		≈ 0.87	MovieLens
Töscher *et al.* [16]	Matrix Factorization		≈ 0.85	Netflix
Wen [17]	Item-Based EM and Sparse SVD		≈ 0.89	Netflix
Bell and Koren [18]	Pearson-r and neighbour selection		≈ 0.89	Netflix

Table 2.6: Error rates of existing systems and the dataset used for the evaluation

which is significantly below these error rates.

In order to compare the results from the proposed system that is presented in this thesis, the evaluation takes dataset from MovieLens and the MAE into account. Besides the dataset from MovieLens and the calculation by the usage of the MAE, the evaluation also considers a dataset from a survey and the MSE and the RMSE.

2.2.3 Hybrid Recommendation Systems

Hybrid recommendation systems combine more than one technique [74]. They can consider the content-based, the collaborative-filtering approach, and other techniques [75]. This is quite useful, because each approach has some weaknesses, which can be minimized by combining these different approaches. The following paragraphs will present the work which uses hybrid recommendation systems.

Martínez *et al.* [39] propose a hybrid recommendation system that considers content-based and collaborative filtering algorithms. The authors use the SVD that is able to reduce the dimensions of a user-item matrix, which contains the ratings from users on items. The considered user profiles are extracted from a social network. The creation of the user profiles is realized by using the explicit settings. Users are ask to set their preferences explicitly. Besides the explicit creation of the user profiles, the system also logs the behaviour in an implicit manner. The calculation of the similarities is performed by the usage of the *Cosine Similarity.* However, the proposed system of this thesis uses a selection of multiple SotA and newly developed algorithms.

The system from George Lekakos and Petros Caravelas [76] uses a hybrid approach for the creation of movie recommendations. The authors use the *Pearson-r Correlation* for the similarity calculation. After the similarity calculation, the system uses these similarities between the active user and the other users to find the k-nearest neighbours. The prediction calculation uses these neighbours. The content-based predictor is calculated by using the *Cosine Similarity*. The system is able to consider metadata, like cast, directors, writers, producers, and the genre and the plot words. Each movie is represented as a vector and the *Cosine Similarity* computes similarities between these vectors. The used algorithm is an extension of the top-N item-based algorithm, which is presented in [77].

The **A**d**VA**nced **T**elematic search of **A**udiovisual contents by semantic **R**easoning (AVATAR) system from Blanco *et al.* [78] considers collaborative-filtering algorithms and content-based filtering methods. In addition, the proposed system uses its own ontology, as shown in Figure 2.6, to build a TV hierarchy. These kinds of ontologies can be used to structure information [79]. The TV ontology from the AVATAR system is described by the means of Web Ontology Language (OWL) [80].

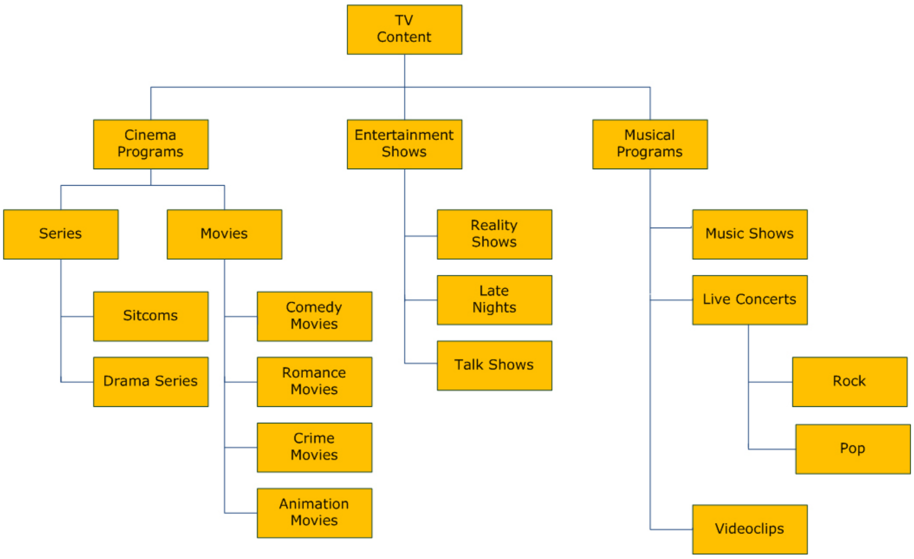


Figure 2.6: AVATAR - Ontology [78]

Figure 2.6 illustrates that the TV content hierarchy contains so-called “superclasses” and “classes”. The lowest unit in this hierarchy is the “class”.

In order to create recommendations, the AVATAR system calculates a Degree of Interest (DOI), which is defined by Equation 2.17 [78].

$$DOI(C_m) = \frac{DOI(C_{m+1})}{1 + \#sib(C_{m+1})} \tag{2.17}$$

$DOI(C_m)$ is the superclass of C_{m+1} and $\#sib(C_{m+1})$ represents the number of siblings of the class C_{m+1}.

The system calculates a matching. This matching takes the DOI and a semantic similarity into account. The matching is calculated by using Equation 2.18 [78]. Equation 2.19 defines the calculation of the semantic similarity [78].

$$match(a, U) = \frac{1}{\#N_U} \sum_{i=1}^{\#N_U} SemSem(a, c_i) \cdot DOI(c_i) \tag{2.18}$$

c_i is the i-th content, which is defined in the user profile P_U. $DOI(c_i)$ represents the level of interest of U regarding c_i. $\#N_u$ represents the total number of programs included in P_U.

SemSem is the semantic similarity, which is described by Equation 2.19 [78]. It uses the hierarchical and the inferential similarity, which are combined by means of a factor $\alpha \in [0, 1]$.

$$SemSem(a, b) = \alpha \cdot SemSem_{Inf}(a, b) + (1 - \alpha) \cdot SemSem_{Hie}(a, b) \quad (2.19)$$

In addition to the proposed ontology, the AVATAR system also uses so-called semantic characteristics, like *hasActor, hasActress, hasTopic, hasTime, hasPlace, etc.* These characteristics permits the system to infer hidden knowledge in the used ontology. The proposed system calculates similarities by using the *Pearson-r correlation.*

In contrast to the AVATAR system, the presented thesis just uses specified metadata, which is sent within the DVB Transport Stream and specified by ETSI [20]. The usage of the metadata from DVB guarantees that each event is enriched with specified information, such as the genre, the title, and so forth. The specified information is used to create the user-item matrix which contains the ratings from users on specified genres. This procedure guarantees that the proposed system is always able to use the information within the user-item matrix for the creation of the recommendations.

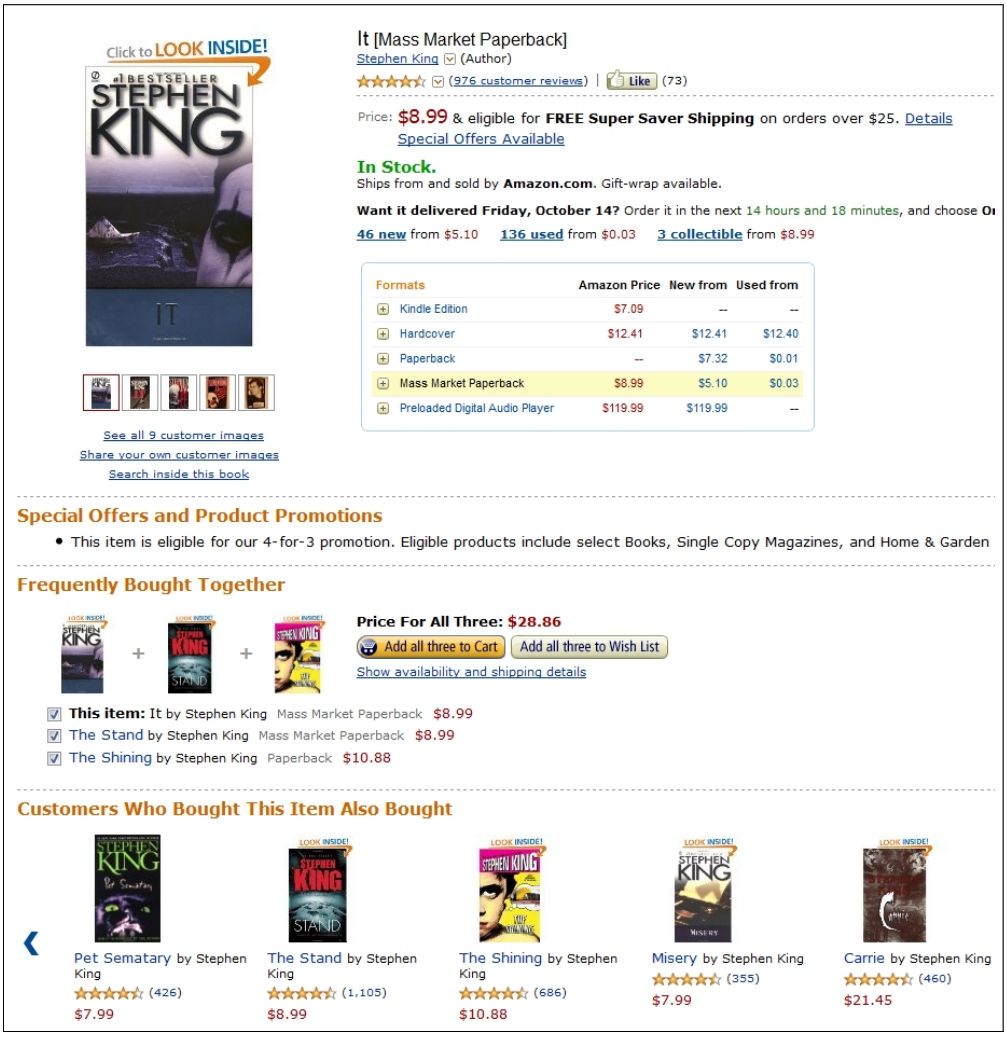
2.3 Presentation of Recommendations

The presentation of recommendations is realized by a lot of online shops, music portals, video portals, etc. The online Internet shop Amazon presents recommendations as well. If a customer selects an article, Amazon presents similar articles. “Customers who bought this also bought...”, as shown in Figure 2.7. In addition, the users are able to see ratings from the related articles, which are based on users’ feedback.

A music portal named LastFM logs the playlists of users. With these playlists LastFM recommends other music tracks and plays them. LastFM offers also the opportunity for finding neighbours who have a similar user profile. Users can browse the neighbours’ playlists and can play tracks from related artists.

YouTube applies recommendation techniques [81]. For example, YouTube offers the opportunity for searching related videos. This feature helps users to have access to related videos without browsing the immense number of available video clips, as shown in Figure 2.8.

The interface from Ardissono *et al.* [82] is able to present recommendations for TV content. It visualizes metadata, like the start time, a category, a title of an event, and the channel. Besides these metadata, the interface also presents recommendations. The grade of the recommendations is presented by smilies, where five smilies represent definite interest and zero smilies no interest. Figure 2.9 shows a screenshot of this interface.



It [Mass Market Paperback]
Stephen King (Author)
(976 customer reviews) | Like (73)

Price: $8.99 & eligible for **FREE Super Saver Shipping** on orders over $25. Details
Special Offers Available

In Stock.
Ships from and sold by **Amazon.com.** Gift-wrap available.

Want it delivered Friday, October 14? Order it in the next 14 hours and 18 minutes, and choose **O**

46 new from $5.10 **136 used** from $0.03 **3 collectible** from $8.99

Formats	Amazon Price	New from	Used from
⊞ Kindle Edition	$7.09	--	--
⊞ Hardcover	$12.41	$12.41	$12.40
⊞ Paperback	--	$7.32	$0.01
⊞ Mass Market Paperback	$8.99	$5.10	$0.03
⊞ Preloaded Digital Audio Player	$119.99	$119.99	--

Special Offers and Product Promotions

- This item is eligible for our 4-for-3 promotion. Eligible products include select Books, Single Copy Magazines, and Home & Garden

Frequently Bought Together

Price For All Three: $28.86

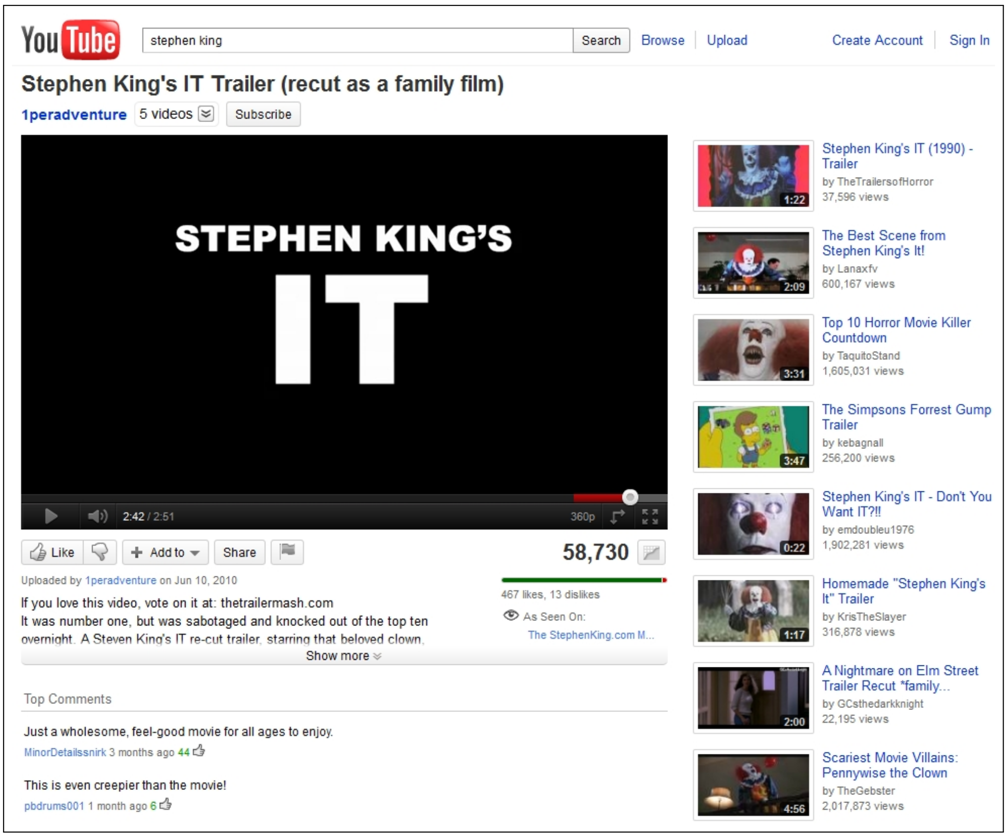
Show availability and shipping details

☑ **This item:** It by Stephen King Mass Market Paperback $8.99
☑ The Stand by Stephen King Mass Market Paperback $8.99
☑ The Shining by Stephen King Paperback $10.88

Customers Who Bought This Item Also Bought

Title	Rating	Price
Pet Sematary by Stephen King	(426)	$7.99
The Stand by Stephen King	(1,105)	$8.99
The Shining by Stephen King	(686)	$10.88
Misery by Stephen King	(355)	$7.99
Carrie by Stephen King	(460)	$21.45

Figure 2.7: Amazon recommendations which are created by the usage of collaborative-filtering techniques



Stephen King's IT Trailer (recut as a family film)

1peradventure

Uploaded by 1peradventure on Jun 10, 2010

If you love this video, vote on it at: thetrailermash.com
It was number one, but was sabotaged and knocked out of the top ten overnight. A Steven King's IT re-cut trailer, starring that beloved clown,

467 likes, 13 dislikes

As Seen On:
The StephenKing.com M...

Top Comments

Just a wholesome, feel-good movie for all ages to enjoy.
MinorDetailssnirk 3 months ago 44

This is even creepier than the movie!
pbdrums001 1 month ago 6

Stephen King's IT (1990) - Trailer
by TheTrailersofHorror
37,596 views

The Best Scene from Stephen King's It!
by Lanaxfv
600,167 views

Top 10 Horror Movie Killer Countdown
by TaquitoStand
1,605,031 views

The Simpsons Forrest Gump Trailer
by kebagnall
256,200 views

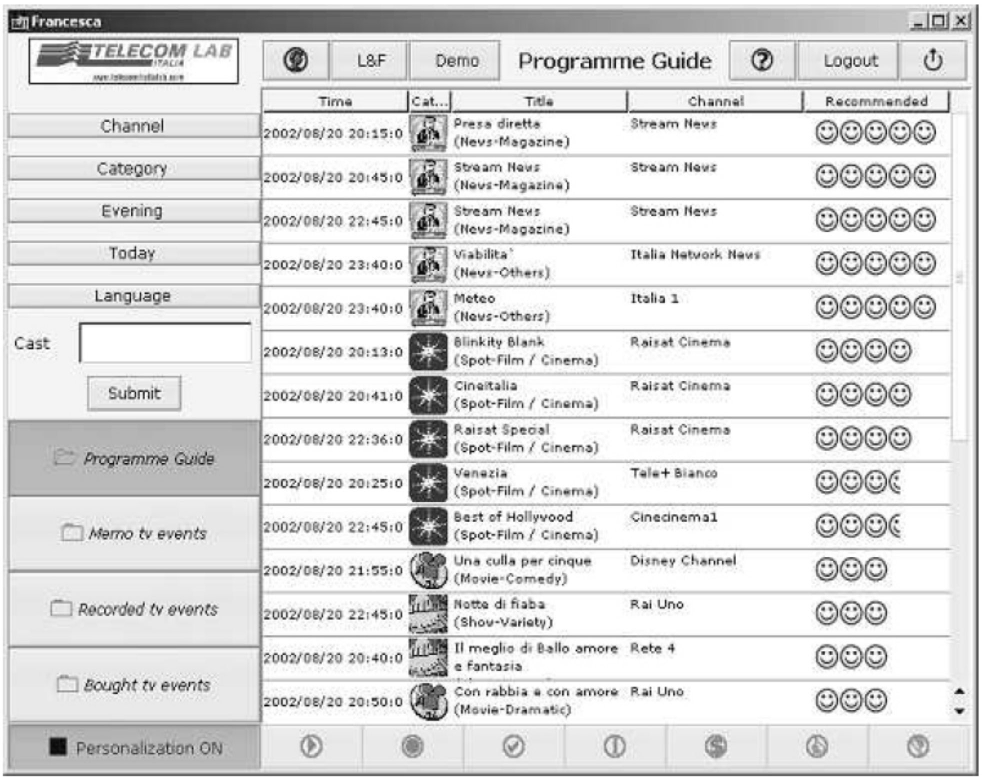
Stephen King's IT - Don't You Want IT?!!
by emdoubleu1976
1,902,281 views

Homemade "Stephen King's It" Trailer
by KrisTheSlayer
316,878 views

A Nightmare on Elm Street Trailer Recut *family...
by GCsthedarkknight
22,195 views

Scariest Movie Villains: Pennywise the Clown
by TheGebster
2,017,873 views

Figure 2.8: YouTube - Related video clips which are based on the selected video clip



Time	Cat...	Title	Channel	Recommended
2002/08/20 20:15:0		Presa diretta (News-Magazine)	Stream News	☺☺☺☺☺
2002/08/20 20:45:0		Stream News (News-Magazine)	Stream News	☺☺☺☺☺
2002/08/20 22:45:0		Stream News (News-Magazine)	Stream News	☺☺☺☺☺
2002/08/20 23:40:0		Viabilita' (News-Others)	Italia Network News	☺☺☺☺☺
2002/08/20 23:40:0		Meteo (News-Others)	Italia 1	☺☺☺☺☺
2002/08/20 20:13:0		Blinkity Blank (Spot-Film / Cinema)	Raisat Cinema	☺☺☺☺
2002/08/20 20:41:0		CineItalia (Spot-Film / Cinema)	Raisat Cinema	☺☺☺☺
2002/08/20 22:36:0		Raisat Special (Spot-Film / Cinema)	Raisat Cinema	☺☺☺☺
2002/08/20 20:25:0		Venezia (Spot-Film / Cinema)	Tele+ Bianco	☺☺☺☺
2002/08/20 22:45:0		Best of Hollywood (Spot-Film / Cinema)	Cinecinema1	☺☺☺☺
2002/08/20 21:55:0		Una culla per cinque (Movie-Comedy)	Disney Channel	☺☺☺
2002/08/20 22:45:0		Notte di fiaba (Show-Variety)	Rai Uno	☺☺☺
2002/08/20 20:40:0		Il meglio di Ballo amore e fantasia	Rete 4	☺☺☺
2002/08/20 20:50:0		Con rabbia e con amore (Movie-Dramatic)	Rai Uno	☺☺☺

Figure 2.9: Recommendation interface from Ardissono *et al.* [82]

2.4 Summary

This chapter presented the related work of recommendation systems. At the beginning it introduced the reader to the topic of user profiling. Secondly it presented the different filtering techniques. Since the main task of this thesis focuses on collaborative-filtering techniques, the content-based filtering was described briefly. It showed the results of existing recommendation systems and the used datasets for the evaluation of these existing systems. In addition this chapter tackled the hybrid recommendation systems which include more than one filtering technique. Finally, the chapter introduced the reader to different manners of the presentation of recommendations.

Chapter 3

Methodology

This chapter tackles the methodology of the thesis. Firstly it presents an environment which represents a home scenario. This home environment was used for the implementation. Secondly it presents the datasets which are used to evaluate the proposed dynamic multi-algorithm collaborative-filtering system. Thirdly it presents the techniques for the creation of the user profiles. Additionally the chapter presents the used filtering techniques and describes the calculation of the predictions and the error rates. Besides these aspects, the used metadata is described as well. Finally the evaluation of the proposed system is described.

3.1 Home Environment

The proposed dynamic multi-algorithm collaborative-filtering system of this thesis is partially implemented in a so-called home environment. Figure 3.1 illustrates a possible scenario of a home environment.

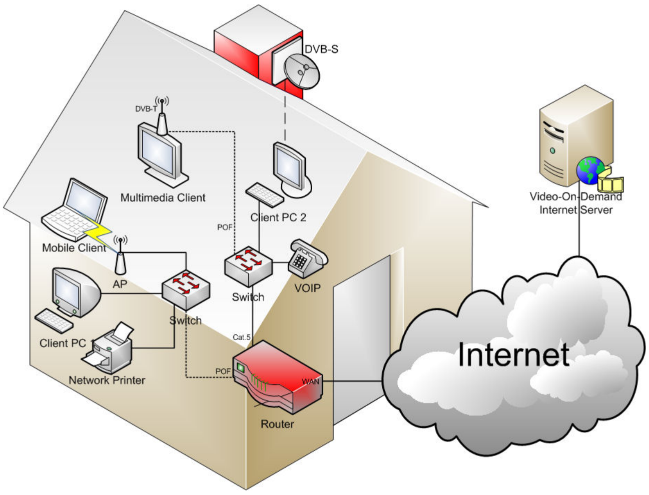


Figure 3.1: HomeVision - Media Convergent Service Environment

The centre of this scenario is a router. The router is responsible for managing the user profiles that are created in an implicit and/or explicit manner. The set preferences are stored in an Extensible Markup Language (XML) [83] that is saved on the router. The router also offers access to the Internet. Therefore the interface, which is presented in Chapter 6, can access data from YouTube. The updating of the user profiles can easily be realized by using Hypertext Preprocessor (PHP).

3.2 Datasets - User-Item Matrices

In order to evaluate the proposed system, the presented thesis considers two datasets.

One dataset is the result of a survey which was undertaken at the THM. Users were asked to set their preferences by rating genres. These genres are specified by an ETSI standard for Service Information [20]. Table 3.1 presents the results of this survey which are used to build the so-called user-item matrix. Ten users were asked to set their likings on specified DVB genres by setting a rating between 0 and 5. 0 represents no interest in the selected genre and 5 represents definite interest in the selected genre. In this user-item matrix a genre represents an item. The presented results from this survey use these specified main genres, which are shown as I1-I12 in Table 3.1. Table 3.2 illustrates which item belongs to which genre. This mentioned dataset represents a small group of users.

These preferences are saved as user profiles, which can be created in an implicit and an explicit manner.

In addition to the dataset from the survey, the presented thesis also considers a dataset from MovieLens, which includes ratings from 943 users on 1682 movies. This dataset is used because it is well-known [84] and the experiments shall compare the achieved results with the results from other researchers, such as [7, 10, 14, 15, 40]. The existing approaches from other researchers, who evaluated their system with the dataset from MovieLens are presented in Section 2.

	U1	**U2**	**U3**	**U4**	**U5**	**U6**	**U7**	**U8**	**U9**	**U10**
I1	5	5	5	3	1	5	5	5	5	5
I2	4	4	4	3	1	5	2	4	5	5
I3	3	3	2	1	1	0	0	0	1	2
I4	5	0	1	1	4	1	0	5	5	4
I5	0	4	0	0	0	0	0	0	0	0
I6	2	0	0	0	3	1	2	3	3	2
I7	1	2	1	3	0	1	4	3	3	4
I8	3	4	3	3	0	3	1	0	4	4
I9	2	4	5	3	5	5	3	5	4	5
I10	2	2	4	0	4	4	0	2	3	4
I11	1	3	1	3	0	0	3	2	3	0
I12	1	3	2	3	5	0	0	2	3	0

Table 3.1: User-item matrix which includes the ratings from ten users on 12 genres (items) that is created by taking the results from the survey into account

3.3 User Profiling

The creation of the recommendations can be based on user profiles. These user profiles contain data which describe the preferences of a user in more detail. User profiles can contain data like gender, age, education, preferences, and so forth. Since the proposed system deals with movie recommendations, the presented thesis focuses on the preferences, which are represented with a value. This value is in the range [0;5], where 0 represents no interest and 5 definite interest in a selected item. Most of the existing systems in the field of movie recommendation systems which are based on collaborative-filtering techniques use this approach.

However, the presented system logs the viewing behaviour of an individual user, the active user, and creates a RI which will be saved in an XML file. The implicit creation of user profiles is described in Section 4.1.2. The explicit user

Item	**Genre**
I1	Movie/Drama
I2	News/Current Affairs
I3	Show/Game Show
I4	Sports
I5	Children's/Youth programmes
I6	Music/Ballet/Dance
I7	Arts/Culture
I8	Social/Political issues/Economics
I9	Education/Science/Factual topics
I10	Leisure hobbies
I11	Other
I12	Undefined Content

Table 3.2: Assignment of the appreciations between the item number and the genres specified by ETSI

profile is created by the usage of an developed interface, the PPG. Within this PPG users are able to set their preferences, which is briefly described in Section 4.1.1. The features of the PPG are presented in Chapter 6.

3.4 Filtering Techniques

In order to use the created user profiles for the recommendation creation, filtering techniques are used. The filtering techniques can be divided into two main approaches, the content-based filtering and the collaborative-filtering approach. These two different techniques will be described below. Since the main part of this thesis deals with the accuracy improvement by the usage of collaborative-filtering techniques, the content-based approach will be described only briefly.

3.4.1 Content-Based Filtering

Content-based filtering uses metadata, which describes the content in more detail [85, 86]. For example, a video clip from YouTube has a title, a description, a duration, a category, and so forth. The content-based filtering approach uses this information for the filtering. For example, users are able to search for a video clip on YouTube by entering a title into a search bar. YouTube will search for this title within its database and present the search results to the user.

However, the presented PPG uses the content-based filtering approach for the creation of the recommendations. It parses the DVB Service Information and extract data, such as title of the events, genre, subgenre, duration, and so forth. Besides the extraction of the metadata from DVB, the presented PPG also extracts data from YouTube. An overview of the used metadata is presented in Section 3.5.

3.4.2 Collaborative Filtering

In contrast to the content-based filtering approach, the collaborative-filtering approach uses data from a community for the creation of the recommendations. As mentioned above, the experiments of this thesis use the results from a survey, which represents a small dataset and a dataset from MovieLens, which represents a large dataset.

However, the presented dynamic multi-algorithm collaborative-filtering system uses several filtering algorithms, a method for the calculation of predictions and an error analysis. In addition the presented system is able to use the k-nearest neighbour approach. These mentioned elements are briefly described in the following sections.

3.4.2.1 Filtering Algorithms

The proposed dynamic multi-algorithm collaborative-filtering system uses filtering algorithms, which are responsible for calculating similarities between users or items. If the recommendation engine calculates similarities between users, the user-based approach is used [87, 88]. The calculation of the similarities between items is called item-based [87–89]. Many publications which deal with collaborative-filtering systems use one of the following algorithms:

- Pearson-r Correlation
- Spearman Rank Correlation
- Cosine Similarity
- Adjusted Cosine Similarity

These algorithms are described in Section 2.2.2.1 in more detail. The algorithms, except the *Cosine Similarity*, deliver a value between -1 and 1, where 1 represents full similarity between two objects, 0 represents no similarity, and -1 represents full negative similarity between two objects. The *Cosine Similarity* delivers a value between 0 and 1, while 0 represents no similarity and 1 full similarity between two objects. Besides these well-known algorithms, newly developed algorithms are presented too. These novel algorithms, which overcome researched weaknesses of the mentioned existing algorithms, are presented in Section 4.3.1.

3.4.2.2 Prediction

In order to predict an item (an entry within the user-item matrix) a method is needed which is able to calculate predictions. Table 3.3 illustrates the ratings from a user on items. Within this table the user has not set her/his preferences for item numbers 4 and 8. These items shall be predicted. However, the presented thesis uses the *Weighted Sum* approach for the calculation of a prediction. This approach is described in Section 4.3.3 and Equation 2.14 defines the calculation of the predictions. Since the thesis focuses on the improvement of the predictions' accuracy it does not tackle the well-known sparsity problem that refers to a situation that data are lacking [90].

3.4.2.3 Error Rates

The accuracy of the predictions is exploited by the MAE, the MSE, and the RMSE. These error rates compare the prediction of an entry with the original value. In this thesis the original value is represented by rating for

Item	**Rating**
I1	5
I2	4
I3	3
I4	?
I5	1
I6	3
I7	4
I8	?
I9	2
I10	2
I11	5
I12	1

Table 3.3: Example of the ratings from a user on specified genres

a item within the user-item matrix. The equations are described in Section 4.3.4 in more detail. Since the proposed system shall be compared with existing recommendation systems, such as [7,10,11,14–18,40], the presented thesis uses these error rates.

3.4.2.4 K-Nearest Neighbours

As mentioned above, collaborative-filtering systems use data from a community. The k-nearest neighbour approach finds the neighbours within a given threshold - it uses just users or items which are quite "similar" to the active user or item [7,65,66,91,92]. For example, if the community contains 1000 users, the system could use the ten nearest neighbours for further calculations. The evaluation of this thesis proves that the error rates decrease by the usage of the k-nearest neighbour approach. Due to this fact this thesis uses this approach, which is described in Section 4.3.2 in more detail.

3.5 Metadata

3.5.1 Digital Video Broadcast

DVB content is enriched with metadata. These metadata are called Service Information (SI) [20, 93]. The SI is sent within the DVB Transport Stream and is packed in tables. One table contains the metadata, which describes the content in more detail. This table is the Event Information Table (EIT) and contains metadata like:

- title of the events
- genre
- duration
- start time and end time
- start date
- ...

The proposed recommendation system includes a DVB-SI parser, which is able to extract the mentioned metadata from the DVB Transport Stream. The Transport Stream is basically a coding for moving pictures and associated audio [94]. Within the EIT several descriptors can be read. The proposed system uses the extended event descriptor that is a table which is included into the EIT. This table, which is sent within the EIT, contains the above mentioned metadata.

3.5.2 YouTube

YouTube videos are enriched with metadata and can be parsed through an Application Programming Interface (API) from Google. This API is able to extract metadata such as a title, the duration, the description, the video format, and so forth. The presented thesis uses this API from Google and the given methods for the extraction of the metadata. The API is used within the developed PPG that is presented in Chapter 6. It can be downloaded by using the following Uniform Resource Locator (URL): `http://code.google.com/intl/de-DE/apis/youtube/getting_started.html#data_api`.

3.6 Evaluation

The evaluation of this thesis focuses on the accuracy of the predictions by the usage of the proposed dynamic multi-algorithm collaborative-filtering system. The evaluation uses SotA approaches, such as the prediction calculation and the calculation of the error rates. Existing approaches use the MAE or the RMSE for the evaluation of their systems. The presented thesis exploits these error rates and compares the results from existing systems with the proposed system. In order to prove that the proposed system is able to consider small and huge datasets, the presented thesis uses the outcome of the above mentioned survey and the dataset from MovieLens. The evaluation also includes results which compare the traditional prediction calculation with the results that have been achieved by the usage of a prediction truncation. The advantage of the k-nearest neighbour approach is also proved with results. The need for the dynamic selection of the most accurate algorithm completes the evaluation and proves the usefulness of the proposed system.

The following sections describe the accomplished experiments:

- Prediction Truncation
- Without Neighbourhood
- With Neighbourhood
- Dynamic Selection

3.6.1 Prediction Truncation

The usefulness of prediction truncation is presented in Section 5.1. This section compares the error rates by the usage of the dataset from MovieLens. The results have been achieved by calculating the prediction of each entry within the user-item matrix from MovieLens. Each entry has been deleted, the prediction has been calculated by the usage of each algorithm, and the error rates have been calculated. With this technique, each single entry of the user-item matrix is considered.

3.6.2 Without Neighbourhood

In order to prove the usefulness of the proposed dynamic multi-algorithm collaborative-filtering system, the presented thesis accomplishes tests that do not consider this proposed approach nor the k-nearest neighbour approach. These tests consider the dataset from the survey and the dataset from MovieLens. The tests also take the user-based approach and the item-based approach into account.

3.6.2.1 Survey

The tests which consider the dataset from the survey take five different user-item matrices into account. The first user-item matrix uses all ten users. The second user-item matrix was built by using user 1 - user 5. The third test was performed by using user 6 - user 10. The fourth user-item matrix contains the ratings from user 1, user 3, user 5, user 7, and user 9. The last test uses user 2, user 4, user 6, user 8, and user 10. The results of these tests prove that the most accurate algorithm is strongly connected to the used dataset. In addition the results prove that a dynamic choice of the most accurate algorithm reduces the error rate.

The following example shall clarify the procedure of the prediction calculations.

Example

Table 3.1 contains the ratings from ten users, which rated twelve genres. The evaluation process deletes the first entry of this table and calculates the similarities by using each algorithm within the proposed dynamic multi-algorithm collaborative-filtering system, which is presented in Section 4.3 in more detail. If the calculation of the similarities is finished, the prediction of the deleted entry can be calculated by using the *Weighted Sum* approach, which is defined by Equation 2.14. This calculated prediction is used to calculate an error rate, such as the MAE, the MSE, and the RMSE. This procedure will be realized with every entry from every user. The result is an error rate which is built by using every error rate from every entry and every user.

3.6.2.2 MovieLens

The results that have been achieved by using the dataset from MovieLens prove that the newly developed algorithms are able to reduce the error rates comparing to SotA collaborative-filtering algorithms. The results have been achieved by calculating the predictions and error rates for every single entry within the dataset from MovieLens, as described above.

3.6.2.3 Performance

Besides the error rates, the thesis also presents performance results. The measurement of the performance has been accomplished by the usage of the dataset from MovieLens, the dataset from the survey, and a simulated user-item matrix using several numbers of entries. The results present the duration of the calculations.

3.6.3 With Neighbourhood

In order to prove the usefulness of the k-nearest neighbour approach, experiments are undertaken which take this approach into account. The results prove that the usage of the k-nearest neighbour approach is able to improve the prediction accuracy. Besides this improvement, the experiments also prove that the calculation duration is significantly lower compared to the results that do not use the nearest neighbour approach.

The experiments consider the dataset from the survey and the dataset from MovieLens.

3.6.4 Dynamic Selection

The usefulness of the proposed dynamic multi-algorithm collaborative-filtering system is proved by the comparison between the above mentioned results with the results that take the proposed system into account. The experiments consider the dataset from the survey and the dataset from MovieLens.

The proposed dynamic selection of the most accurate algorithm is split into several parts. The first step of the proposed system selects the active user or item. The second step of the process selects an algorithm and calculates the similarities between the active user/item and the entire users/items that are included into the user-item matrix. The calculated similarities are used to build a new user-item matrix that only contains the active user/item and the k-nearest users/items. The third step predicts every entry of the active user/item and compares the predictions with the original entries such that the error rates can be calculated. The second and the third third step is undertaken with every algorithm that is included in to the proposed system. At the end of the entire process the system proposes the most accurate collaborative-filtering algorithm which is strongly connected to the active user/item and its neighbourhood.

The results of the evaluation prove that the proposed system improves the prediction accuracy significantly compared to the above mentioned approaches. Besides this fact, the comparison between the proposed system and existing recommendation systems also prove this improvement.

3.7 Summary

This chapter introduced the reader to the methodology of the thesis. Firstly it presented the used home environment. Secondly it presented the datasets which are used to evaluate the proposed dynamic multi-algorithm collaborative-filtering system. The introducing of the datasets includes a small dataset that is the outcome of an undertaken survey and a large dataset from MovieLens. Besides these topics, this chapter also described the used filtering techniques, such as the content-based filtering and the collaborative filtering. It introduced the calculation of predictions, the used error calculation, and the k-nearest neighbour approach. Additionally it presented the used metadata. Finally it described the accomplished evaluation of the proposed system.

Chapter 4

Recommendation System

The following sections presents the researched and developed recommendation system. It tackles the different kinds of user profiling, which can be realized in an explicit and an implicit manner. In addition it presents the content-based filtering approach and a collaborative filtering system, which includes newly developed algorithms.

In general the proposed system uses metadata, which is sent within the DVB Transport Stream. A DVB Transport Stream contains the video and audio signal as well as Service Information. This Service Information is specified by a standard from ETSI. The Service Information is sent within tables and the used information is part of the EIT. This EIT contains the metadata, which describes the events in more detail, e.g. genre, subgenre, title of the event, duration of the event, long description, and so forth. The ETSI standard for Service Information specifies twelve main genres, as shown in Table 3.2.

Each genre is split into several subgenres, which classifies the genres in

Genre	**Subgenre**
Movie/Drama	Movie/Drama (general) Detective, Thriller Adventure, Western, War Science-Fiction, Fantasy, Horror Comedy Soap, Melodrama Romance Serious, Classical, Religious, Historical Adult Movie/Drama

Table 4.1: Subgenres from the Genre: Movie/Drama which are specified by ETSI

more detail. Table 4.1 presents all subgenres of the genre 'Movie/Drama'.

However, a video or audio recommendation system is mainly split into three parts.

- User Profiling
- Filtering
- Presentation of the recommendations

This chapter describes the first two parts. Section 4.1 describes the user profiling approach in more detail. It considers the implicit creation and the explicit creation of user profiles. Section 4.2 briefly introduces the reader to the content-based filtering approach. Section 4.3 presents the newly developed and researched dynamic multi-algorithm collaborative-filtering system. It includes SotA collaborative-filtering algorithms and newly developed collaborative-filtering algorithms, which overcome researched weaknesses of existing algorithms. The presentation of the recommendations is realized by a developed PPG, which is presented in Chapter 6.

4.1 User Profiling

In order to generate recommendations, user profiles are needed. These user profiles contain the preferences of individual users. However, the explicit and the implicit creation of the user profiles will be described in the following sections.

4.1.1 Explicit Profiling

The explicit user profile is created by setting preferences. These preferences are saved as ratings on e.g., movies, genres, and so forth. The ratings within the proposed system are saved as a number. In contrast to the implicitly created user profile, which is described in the following section, the explicitly created user profile contains more accurate information [95]. Users are able to set their individual preferences [29], e.g. by setting stars. This approach is quite common and is used by several kinds of applications.

The presented approach uses this kind of setting. Users are able to rate events and genres/subgenres by setting stars. Five stars represents definite interest in the selected event, genre, or subgenre. Users are also able to exclude events, genres, or subgenres from their user profile. Figure 4.1 shows a screenshot of the developed PPG.

Within this page of the PPG, users are able to set their preferences by setting stars. If a user had chosen her/his likings, these preferences will be saved in an XML file. An example of an XML file is shown in Listing 4.1. The number of stars will be converted into a RI. Five stars will be converted

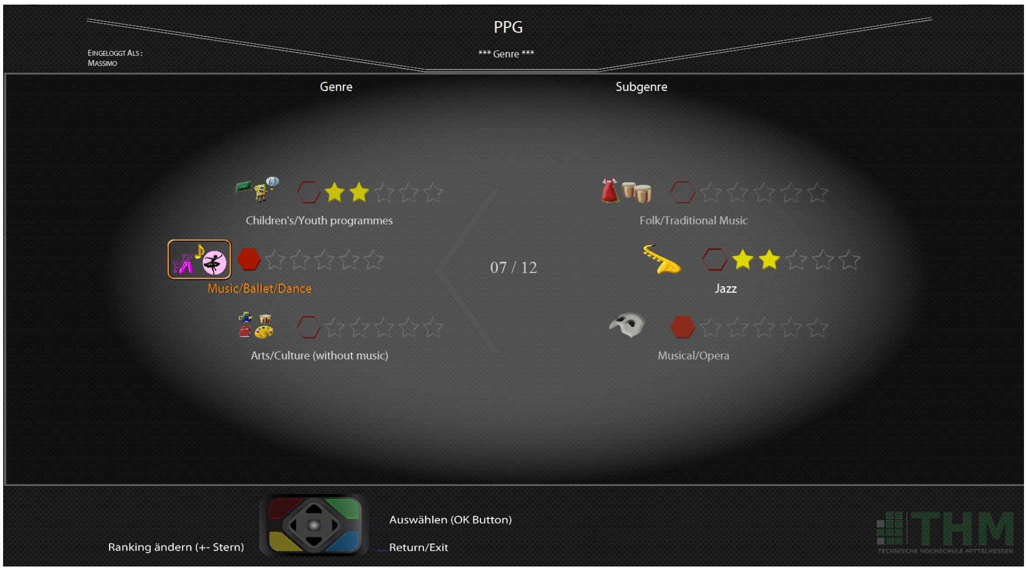


Figure 4.1: PPG - Explicit Settings

into a RI of 1. Four stars will be saved as 0.8, three stars as 0.6, two stars as 0.4, one star as 0.2, no stars as 0, and the stop sign, which represents no interest in the selected genre, subgenre, or event, will be saved as -500.

```xml

<user name="chris">
 <favouritemaingenres id="explicit">
  <genre mgenre="0x10" mRI="0.2"/>
  ...
 </favouritemaingenres>

 <favouritesubgenres id="explicit">
  <genre sgenre="0x11" sRI="0.6"/>
  ...
 </favouritesubgenres>

 <favouriteevents id="explicit">
  <event name="nano" eRI="0.8"></event>
  ...
 </favouriteevents>
</user>
```

Listing 4.1: User Profile XML - implicit

mgenre represents a maingenre, *sgenre* a subgenre. *mRI* is the RI of a maingenre, *sRI* of a subgenre, and *eRI* the RI of an event. The RI is defined in Section 4.1.2.

4.1.2 Implicit Profiling

The viewing behaviour can be logged for the creation of implicit user profiles [21, 29, 49, 53, 96]. This implicitly logged viewing behaviour can be used to enrich the features of a recommendation system. In contrast to the explicitly set preferences, the implicit profiling also takes the real behaviour into account. For instance, if a user does not set the genre "education" in an explicit manner, but he/she watches this genre quite often, the implicit profiling will recognize it and add this genre to the user profile. The implicit creation of a user profile also overcomes the problem that users tend not to provide much input in feedback [97]. The implicit creation of the user profiles also overcomes the problem that the information need of users is vague [98]. However, published research work proves the strong correlation between spending time on a single view and the importance of this single view [99]. Due to this fact, the following equations exploit the watching duration of a user and save it to the user profile. They use the sent DVB Service Information to generate a RI, which is in the range [0;1], where 0 represents no interest and 1 represents definite interest.

The following equations are responsible for the creation of user profiles. Since the topic of the thesis focuses on the prediction accuracy of collaborative-filtering algorithms, the equations have not been researched in detail. They shall just show a simple manner for the creation of an implicit user profile and the combining of explicitly and implicitly created user profiles.

4.1.2.1 Recommendation Index - General

Equation 4.1 logs the duration of a genre, a subgenre, or an event and calculates a RI, which will be saved in the implicit user profile for the active user.

$$RI_{implicit} = \sum_{i=1}^{\infty} \frac{t_w(i)}{t_d} \tag{4.1}$$

$RI_{implicit} = [0; 1]$, $t_w = [0; t_d]$, $t_d \geq 0$

t_d represents the duration the user watched television. $RI_{implicit}$ is the value of the RI from a genre, subgenre, or an event which has been calculated. The variable $t_w(i)$ represents the duration the user watched the genre, subgenre, or an event, where i is a counter.

With this counter the presented equation is able to calculate the RI of the watched genre, subgenre, or event over a period of time (t_d). Basically the recommendation system sums the time, the active user watches a genre, a subgenre or an event. This equation is also able to take channel switches into account.

Example:

A user watches a specific movie, like “It”. During the commercial break the user switches to another channel. After a period of time the user switches back to the movie. Equation 4.1 takes this behaviour into account. It sums the time of the watching period before the user switches to another channel

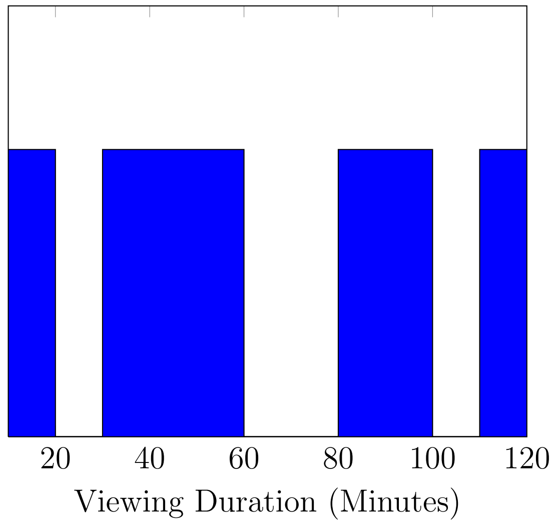


Figure 4.2: Recommendation Index - General

and the time of the watching period after the user switches back to the movie “It”. Figure 4.2 clarifies this behaviour. The blue bars represents the watching time. This figure illustrates that a user watched an event that lasted 120 minutes in total. During this time, the user changed the channel three times. The user always switched back to the event, so that the user watched the event for 80 minutes, while it takes 120 minutes in total. The RI for this event would be $0.\overline{6}$.

4.1.2.2 Recommendation Index - Average

The implicit user profile shall reflect the viewing behaviour of a user. This process is a learning process. The longer a user watches TV, the more accurate the user profile will be. With the intention of guaranteeing that the value of the RI becomes more and more accurate, Equation 4.2 is used to calculate the average.

$$\overline{RI}_{average} = \frac{1}{n}\sum_{k=1}^{n} RI_{implicit}(k) \tag{4.2}$$

The $\overline{RI}_{average}$ is the average value of the RI of a genre, a subgenre, or an event. The expression $RI_{implicit}(k)$ is the value of the RI of one genre, subgenre, or event. The variable (k) is the counter of this genre, subgenre or event and n is the counter of the measurements.

Example:

The user watched the genre movie/drama on Monday. This results in a RI of *0.6*. On Saturday the user watched the genre movie/drama again and the RI for this day is *0.8*. In this case n=2. The average of these RIs is *0.7*, which is shown by Equation 4.3.

$$\overline{RI}_{average} = \frac{0.6 + 0.8}{2} = 0.7 \tag{4.3}$$

The usage of this procedure guarantees that the “real” viewing behaviour is exploited, because the calculation of the RI takes each watching session into account. For example, a user likes to watch a particular soap. The first episode of this soap lasts 60 minutes in total. For some reason the user misses the first twenty minutes of this first episode. The recommendation system would calculate a RI of $0.\overline{6}$ for it. The user watches the second episode for 60 minutes. The RI for this event will be increased by the usage of Equation 4.2. But this equation also will work the other way round. For example, if a user does not like a particular genre, but the TV is on while the user takes a phone call, the RI would be calculated. But if the user watches this genre again and does not spend much time on watching, the RI will be decreased. Figure 4.3 illustrates the possible calculated RIs and the RI after the averaging.

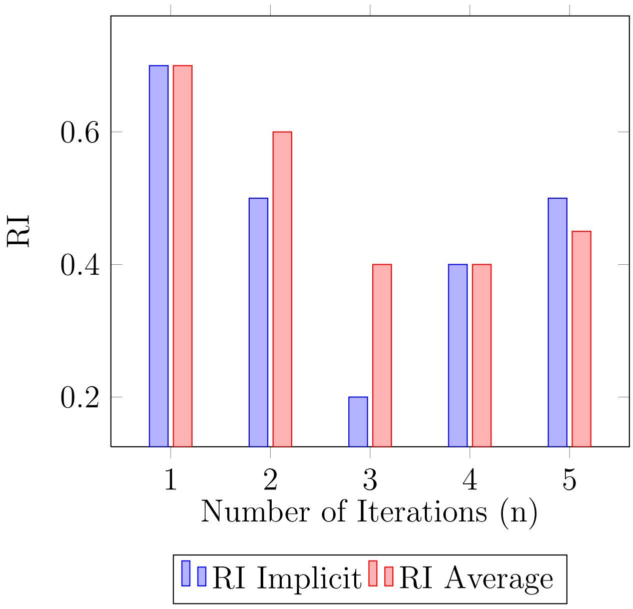


Figure 4.3: Recommendation Index - Average

4.1.2.3 Recommendation Index - Adjustment

Equation 4.1 and Equation 4.2 are responsible for creating the RIs for events, genres, and subgenres in an implicit manner. The above described procedure has one main problem. If the user does not like a particular event, genre, or subgenre, a RI is nonetheless calculated for them even though the user never watches this particular event, genre, or subgenre again, and the RI would never decrease.

Equation 4.4 has been developed to overcome this problem. This equation decreases the RI step by step over time.

$$RI_{adjust} = \overline{RI}_{average} \cdot e^{-(\frac{1}{4})} \tag{4.4}$$

Figure 4.4 shows the RI adjustment. The value of the RI decreases every week statically. After eight weeks the RI is under 0.15, which will be rounded down to zero. This procedure guarantees that the RI will be decreased if a user never watches an event, a genre, or a subgenre again.

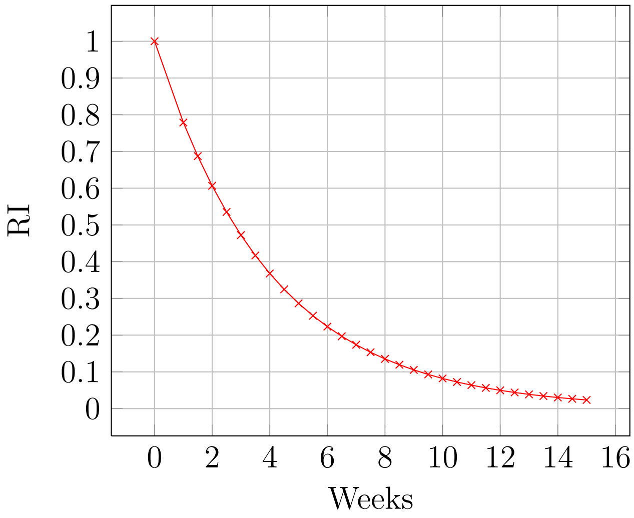


Figure 4.4: Recommendation Index - Adjustment

4.1.2.4 Recommendation Index - Mix

In order to combine events, genres, and subgenres as well as the implicitly and explicitly created user profiles, Equation 4.5 has been developed. Since users feel more comfortable with the explicit system, which is shown by [44], the equation multiplies the explicitly created RI with a factor of two. The basis configuration of the developed PPG presents the recommendations which are based on the explicit user profile. Users are also able to switch to the recommendations, which are generated by considering the implicitly created user profile.

$$RI_{mix} = \frac{RI_{adjust} + RI_{explicit} \cdot 2}{3} \tag{4.5}$$

4.2 Content-Based Filtering

Since this thesis focuses on the accuracy improvement of predictions by the usage of collaborative-filtering algorithms, this section describes the content-based approach only briefly.

In order to offer the opportunity to use the content-based approach, the presented recommendation system includes a developed SI parser which is able to extract metadata from a DVB Transport Stream. This parser extracts data from the EIT, which contains data like title of the events, genres and subgenres, duration, and so forth. The content-based approach uses the DVB-SI and the created user profiles for the creation of recommendations. Besides the SI data, the developed recommendation system uses the API from YouTube for the extraction of metadata from it.

The recommendations are based on the implicitly logged viewing behaviour and/or the explicitly set preferences. Since the recommendation system saves a RI which represents the preference, the recommendation system calculates the RI by taking the approach that is described in the following section into account.

4.2.1 Recommendation Index - Final

An event which is broadcast by DVB is enriched with metadata, such as the title of the event, the genre, the subgenre, and so forth. The title of an event is the most significant description. If a user wants to find an event, she/he will search for the title of it. The genres which are broadcast by DVB are split into several subgenres as shown by Table 4.1. Therefore the subgenres describe the genres in more detail. The following equations take these factors into account.

Basic Scenario:

A user searches for recommendations, which are based on her/his user profile. The user profile contains several RIs for events, genres, and subgenres. The DVB-SI parser extracts the metadata from the DVB Transport Stream, which contains data of the scheduled events that are broadcast currently or in the near future.

Scenario 1:

A title , the genre, and the subgenre of an event from the extracted scheduled information is part of the user profile of the currently logged-in user. Equation 4.6 will be used to calculate the RI for this event.

$$RI = \frac{RI_{event} \cdot 3 + RI_{subgenre} \cdot 2 + RI_{genre}}{6} \tag{4.6}$$

Scenario 2:

If a title of an event and genre from the scheduled information is part of the user profile from the current user, Equation 4.7 will be used to calculate the

RI.

$$RI = \frac{RI_{event} \cdot 3 + RI_{genre}}{4} \tag{4.7}$$

Scenario 3:

If only the title of an event from the scheduled information is part of the user profile from the current user, Equation 4.8 will calculate the RI.

$$RI = RI_{event} \tag{4.8}$$

Scenario 4:

If the scheduled information's event cannot be found in the user profile of the current user, but the subgenre of this event is part of the user profile, the RI calculation is defined as:

$$RI = \frac{RI_{subgenre} \cdot 2 + RI_{genre}}{3} \tag{4.9}$$

Scenario 5:

If only the event's genre of the scheduled information is part of the user profile from the currently logged-in user, Equation 4.10 calculates the RI for this event.

$$RI = RI_{genre} \tag{4.10}$$

However, Chapter 6 describes the entire developed features of the recommendation system in more details which use the above described creation of the RI.

4.3 Dynamic Multi-Algorithm Collaborative-Filtering System

This section introduces the reader to the proposed dynamic multi-algorithm collaborative-filtering system. This system includes SotA algorithms, which are presented in Section 2.2.2.1. These algorithms are the basis for the newly developed algorithms, which try to overcome researched weaknesses of these algorithms. The newly developed algorithms are presented in Section 4.3.1. The mentioned algorithms are used for finding the most adequate algorithm for the dynamically given user-item matrix.

This section is organized as follows: Section 4.3.1 presents these newly developed algorithms that overcome the researched weaknesses of the well-known ones. Besides the algorithms, the used k-nearest neighbour approach will be presented in Section 4.3.2. Section 4.3.3 briefly describes the calculation of the predications. Within this section the prediction truncation is described as well. The calculation of the errors are presented in Section 4.3.4. Section 4.3.5 finally presents the novel approach of the proposed dynamic multi-algorithm collaborative-filtering system.

4.3.1 Newly Developed Collaborative-Filtering Algorithms

In this section the newly developed and researched algorithms are presented. These algorithms take the researched weaknesses into account and overcome them.

4.3.1.1 Absolute Correlation

The *Absolute Correlation* overcomes the researched weakness of the *Pearson-r Correlation*. The linear correlation between two vectors $v_1 = \{0, 1, 2, 3\}$ and $v_2 = \{10, 11, 12, 13\}$ is 1 if the calculation is realized by using the *Pearson-r Correlation*. The *Absolute Correlation* overcomes this problem by multiplying the result of the *Pearson-r Correlation* with a factor. The used factor is the ratio between the magnitudes of the considered vectors. This magnitude is defined as: $\|\vec{v}\| = \sqrt{v_1^2 + ... + v_n^2}$ [100]. If the system calculates the similarities between items, the factor is defined by Equation 4.12 and Equation 4.13. Equation 4.11 defines the calculation of the similarities between items. Equation 4.15 and Equation 4.16 define the factor by using the user-item approach. Equation (4.14) defines the similarity calculation between users. The result of the *Absolute Correlation* is in the range [-1;+1], where -1 represents a full negative correlation, +1 a full positive correlation, and 0 no correlation between the vectors.

$$AC_{sim}(i,j) = \frac{\sum_{\mathrm{u}\in\mathrm{U}}(R_{u,i}-\overline{R}_i)(R_{u,j}-\overline{R}_j)}{\sqrt{\sum_{\mathrm{u}\in\mathrm{U}}(R_{u,i}-\overline{R}_i)^2}\sqrt{\sum_{\mathrm{u}\in\mathrm{U}}(R_{u,j}-\overline{R}_j)^2}} \cdot abs_{ib} \tag{4.11}$$

$AC_{sim}(i,j)$ is the correlation between item i and item j. $R_{u,i}$ is the rating from user u of item i. $R_{u,j}$ is the rating from user u of item j. $\overline{R}_i$ is the average of the ratings from item i and $\overline{R}_j$ is the average of the ratings from item j. $u \in U$ is the summation of the users who rated both items i and j.

$$abs_{ib} = \frac{\left\|\vec{i}\right\|}{\left\|\vec{j}\right\|}, if \left\|\vec{i}\right\| < \left\|\vec{j}\right\| \tag{4.12}$$

$$abs_{ib} = \frac{\left\|\vec{j}\right\|}{\left\|\vec{i}\right\|}, if \left\|\vec{j}\right\| < \left\|\vec{i}\right\| \tag{4.13}$$

$$AC_{sim}(u,v) = \frac{\sum_{\mathrm{i}\in\mathrm{I}}(R_{u,i}-\overline{R}_u)(R_{v,i}-\overline{R}_v)}{\sqrt{\sum_{\mathrm{i}\in\mathrm{I}}(R_{u,i}-\overline{R}_u)^2}\sqrt{\sum_{\mathrm{i}\in\mathrm{I}}(R_{v,i}-\overline{R}_v)^2}} \cdot abs_{ub} \tag{4.14}$$

$AC_{sim}(u,v)$ is the correlation between user u and user v. $R_{u,i}$ is the rating from user u of item i. $R_{v,i}$ is the rating from user v of item i. $\overline{R}_u$ is the average of the ratings from user u and $\overline{R}_v$ is the average of the ratings from user v. $i \in I$ is the summation of the items that are rated by the users u and v.

$$abs_{ub} = \frac{\|\vec{u}\|}{\|\vec{v}\|}, if\ \|\vec{u}\| < \|\vec{v}\| \tag{4.15}$$

$$abs_{ub} = \frac{\|\vec{v}\|}{\|\vec{u}\|}, if\ \|\vec{v}\| < \|\vec{u}\| \tag{4.16}$$

4.3.1.2 Absolute Rank Correlation

The *Absolute Rank Correlation* takes the researched weakness of the *Spearman Rank Correlation* into account. It multiplies the result of the *Spearman Rank Correlation* with a factor. This factor is the magnitude between the two considered vectors. Equation 4.18 and Equation 4.19 define the factor if the item-based approach is used. Equation 4.21 and Equation 4.22 define the factor if the user-based approach is used. Equation 4.17 is used, if the correlation between items shall be calculated and Equation 4.20 defines the calculation, if the user-based approach is taken into account. The results of the *Spearman Rank Correlation* are in the range [-1;+1], where -1 represents a full negative relationship, +1 a full positive relationship, and 0 no relationship between the vectors.

$$ARC_{sim}(i,j) = \frac{\sum_{\mathrm{u \in U}}(Rg_{u,i} - \overline{Rg}_i)(Rg_{u,j} - \overline{Rg}_j)}{\sqrt{\sum_{\mathrm{u \in U}}(Rg_{u,i} - \overline{Rg}_i)^2}\sqrt{\sum_{\mathrm{u \in U}}(Rg_{u,j} - \overline{Rg}_j)^2}} \cdot abs_{rank-ib} \tag{4.17}$$

$Rg_{u,i}$ is the ranked rating from user u of item i. $Rg_{u,j}$ is the ranked rating from user u of item j. $\overline{Rg}_i$ is the average of the ranked ratings from item i and $\overline{Rg}_j$ is the average of the ranked ratings from item j. $u \in U$ is the summation of the users who rated both items i and j. $abs_{rank-ib}$ is the ratio of the magnitudes from the vectors $\vec{i_g}$ and $\vec{j_g}$.

$$abs_{rank-ib} = \frac{\left\|\vec{i_g}\right\|}{\left\|\vec{j_g}\right\|}, if \left\|\vec{i_g}\right\| < \left\|\vec{j_g}\right\| \tag{4.18}$$

$$abs_{rank-ib} = \frac{\left\|\vec{j_g}\right\|}{\left\|\vec{i_g}\right\|}, if \left\|\vec{j_g}\right\| < \left\|\vec{i_g}\right\| \tag{4.19}$$

$$ARC_{sim}(u,v) = \frac{\sum_{\mathrm{i}\in\mathrm{I}}(Rg_{u,i} - \overline{Rg_u})(Rg_{v,i} - \overline{Rg_v})}{\sqrt{\sum_{\mathrm{i}\in\mathrm{I}}(Rg_{u,i} - \overline{Rg_u})^2}\sqrt{\sum_{\mathrm{i}\in\mathrm{I}}(Rg_{v,i} - \overline{Rg_v})^2}} \cdot abs_{rank-ub} \tag{4.20}$$

$Rg_{u,i}$ is the ranked rating from user u of item i. $Rg_{v,i}$ is the ranked rating from user v of item i. $\overline{Rg_u}$ is the average of the ranked ratings from user u and $\overline{Rg_v}$ is the average of the ranked ratings from user v. $i \in I$ is the summation of the items rated by both users u and v. $abs_{rank-ub}$ is the ratio of the magnitudes from the vectors $\vec{u_g}$ and $\vec{v_g}$.

$$abs_{rank-ub} = \frac{\|\vec{u_g}\|}{\|\vec{v_g}\|}, if \|\vec{u_g}\| < \|\vec{v_g}\| \tag{4.21}$$

$$abs_{rank-ub} = \frac{\|\vec{v_g}\|}{\|\vec{u_g}\|}, if\ \|\vec{v_g}\| < \|\vec{u_g}\| \tag{4.22}$$

4.3.1.3 Absolute Original Rank Correlation

The *Absolute Original Rank Correlation* calculates the *Spearman Rank Correlation* and multiplies the result with the *abs* factor. The *abs* factor is built by using the non-ranked values and is defined by Equation 4.12 and Equation 4.13 if the item-based approach is used. Equation 4.15 and Equation 4.16 defines the factor if the user-based approach is used. The calculation of the correlation between items is defined by Equation 4.23. Equation 4.24 takes the user-based approach into account. The results of the *Absolute Original Rank Correlation* are in the range [-1;+1], where -1 represents a full negative relationship, +1 a full positive relationship, and 0 no relationship between the vectors.

$$AORC_{sim}(i,j) = \frac{\sum_{\mathrm{u}\in\mathrm{U}}(Rg_{u,i}-\overline{Rg}_i)(Rg_{u,j}-\overline{Rg}_j)}{\sqrt{\sum_{\mathrm{u}\in\mathrm{U}}(Rg_{u,i}-\overline{Rg}_i)^2}\sqrt{\sum_{\mathrm{u}\in\mathrm{U}}(Rg_{u,j}-\overline{Rg}_j)^2}} \cdot abs_{ib} \tag{4.23}$$

$Rg_{u,i}$ is the ranked rating from user u of item i. $Rg_{u,j}$ is the ranked rating from user u of item j. $\overline{Rg}_i$ is the average of the ranked ratings from item i and $\overline{Rg}_j$ is the average of the ranked ratings from item j. $u \in U$ is the summation of the users who rated both items i and j. abs_{ib} is the ratio of the magnitudes from the vectors $\vec{i_g}$ and $\vec{j_g}$.

$$AORC_{sim}(u,v) = \frac{\sum_{\mathrm{i \in I}}(Rg_{u,i} - \overline{Rg_u})(Rg_{v,i} - \overline{Rg_v})}{\sqrt{\sum_{\mathrm{i \in I}}(Rg_{u,i} - \overline{Rg_u})^2}\sqrt{\sum_{\mathrm{i \in I}}(Rg_{v,i} - \overline{Rg_v})^2}} \cdot abs_{ub} \tag{4.24}$$

$Rg_{u,i}$ is the ranked rating from user u of item i. $Rg_{v,i}$ is the ranked rating from user v of item i. $\overline{Rg_u}$ is the average of the ranked ratings from user u and $\overline{Rg_v}$ is the average of the ranked ratings from user v. $i \in I$ is the summation of the items rated by both users u and v. abs_{ub} is the ratio of the magnitudes from the vectors $\vec{u_g}$ and $\vec{v_g}$.

4.3.1.4 Cosine Co-Rated Similarity

Since the *Cosine Similarity* does not take the co-rated approach into account, the *Cosine Co-Rated Similarity* has been developed. It considers just the co-rated objects. The co-rated approach is described in Section 2.2.2.1.1 and Table 2.1 clarifies this behaviour. Equation 4.25 performs the similarity calculation if the item-based approach is used. The calculation of the similarities by taking the user-based approach into account is defined by Equation 4.26. The results are in the range [0;1], where 1 represents full similarity, and 0 no similarity between the angle of the two vectors.

$$CCS_{sim}(i,j) = \frac{\sum_{\mathrm{u \in U}}(R_i \cdot R_j)}{\sqrt{\sum_{\mathrm{u \in U}}(R_i)^2} \cdot \sqrt{\sum_{\mathrm{u \in U}}(R_j)^2}} \tag{4.25}$$

$CCS_{sim}(i,j)$ is the similarity between item i and item j. R_i is the rating of item i and R_j is the rating of item j. $_{\mathrm{u \in U}}$ is the summation of the users who rated both items i and j.

$$CCS_{sim}(u,v) = \frac{\sum_{\mathrm{i \in I}}(R_u \cdot R_v)}{\sqrt{\sum_{\mathrm{i \in I}}(R_u)^2} \cdot \sqrt{\sum_{\mathrm{i \in I}}(R_v)^2}} \tag{4.26}$$

$CCS_{sim}(u,v)$ is the similarity between user u and user v. R_u is the rating of user u and R_v is the rating of user v. $_{\mathrm{i \in I}}$ is the summation of the items rated by both users u and v.

4.3.1.5 Cosine Rank Similarity

The *Cosine Rank Similarity* combines the approaches of the *Spearman Rank Correlation* and *Cosine Similarity.* It ranks the entries of the user-item matrix and performs the similarity calculation by using the *Cosine Similarity.* Equation 4.27 is used if the item-based approach is used and Equation 4.28 defines the similarity calculation by considering the user-based approach. The results are in the range [0;1], where 1 represents full similarity, and 0 no similarity between the angle of the two vectors.

$$CRS_{sim}(i,j) = \frac{\vec{i_g} \cdot \vec{j_g}}{\left\|\vec{i_g}\right\| \cdot \left\|\vec{j_g}\right\|} \tag{4.27}$$

$\vec{i_g}$ represents the ranks of the object i_g and vector $\vec{j_g}$ represents the ranks of the object j_g. $\left\|\vec{i_g}\right\|$ is the magnitude of vector $\vec{i_g}$ and $\left\|\vec{j_g}\right\|$ is the magnitude of vector $\vec{j_g}$.

$$CRS_{sim}(u,v) = \frac{\vec{u}_g \cdot \vec{v}_g}{\|\vec{u}_g\| \cdot \|\vec{v}_g\|} \tag{4.28}$$

$\vec{u_g}$ represents the ranks of the object u_g and $\vec{v_g}$ represents the ranks of the object v_g. $\|\vec{u}_g\|$ is the magnitude of vector $\vec{u_g}$ and $\|\vec{v}_g\|$ is the magnitude of vector $\vec{v_g}$.

4.3.1.6 Cosine Rank Co-Rated Similarity

The *Cosine Rank Co-Rated Similarity* combines the *Cosine Rank Similarity* and the co-rated approach, which is described in Section 2.2.2.1.1. Equation 4.29 defines the similarity calculation by using the item-based approach and Equation 4.30 takes the user-based approach into account. The results of the *Cosine Rank Co-Rated Similarity* are in the range [0;1], where 1 represents full similarity, and 0 no similarity between the angle of the two vectors.

$$CRCS_{sim}(i,j) = \frac{\sum_{\mathrm{u}\in\mathrm{U}}(Rg_i \cdot Rg_j)}{\sqrt{\sum_{\mathrm{u}\in\mathrm{U}}(Rg_i)^2} \cdot \sqrt{\sum_{\mathrm{u}\in\mathrm{U}}(Rg_j)^2}} \tag{4.29}$$

$CRCS_{sim}(i,j)$ is the similarity between item i and item j. Rg_i is the ranked rating of the item i and Rg_j is the ranked rating of the item j. $_{\mathrm{u}\in\mathrm{U}}$ is the summation of the users who rated both items i and j.

$$CRCS_{sim}(u,v) = \frac{\sum_{\mathrm{i}\in\mathrm{I}}(Rg_u \cdot Rg_v)}{\sqrt{\sum_{\mathrm{i}\in\mathrm{I}}(Rg_u)^2} \cdot \sqrt{\sum_{\mathrm{i}\in\mathrm{I}}(Rg_v)^2}} \tag{4.30}$$

$CRCS_{sim}(u,v)$ is the similarity between user u and user v. Rg_u is the ranked rating of the user u and Rg_v is the ranked rating of the user v. $_{\mathrm{i}\in\mathrm{I}}$ is the summation of the items rated by both users u and v.

4.3.1.7 Absolute Cosine Similarity

The *Cosine Similarity* computes the cosine angle between two vectors. The main issue of this algorithm is the fact that it does not consider the length of the vectors. Assume User 1 rates every item with 1 $v_1 = \{1, 1, 1, 1\}$ and User 2 rates every item with 5 $v_1 = \{5, 5, 5, 5\}$. The cosine angle between these two vectors by using the *Cosine Similarity* is 1, as shown in Figure 4.5, although the ratings differ significantly.

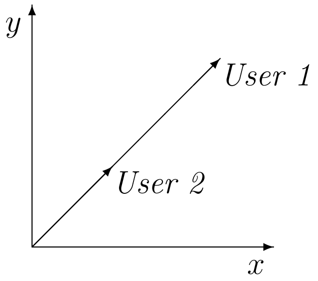


Figure 4.5: Two vectors which are created by the usage of the user's ratings in a two-dimensional room

The *Absolute Cosine Similarity* tackles this problem. It multiplies the result of the *Cosine Similarity* with a factor. This factor is the ratio of the vector magnitudes. The factor is defined by Equation 4.12 and Equation 4.13 if the item-based approach is used. Equation 4.15 and Equation 4.16 is used if similarities between users shall be calculated. The calculation of the *Absolute Cosine Similarity* is defined by Equation 4.31, if the item-based approach is used. Equation 4.32 presents the calculation by considering the user-based approach. The results of the *Absolute Cosine Similarity* are in the range [0;1]. 0 represents no similarity between the two vectors and 1 full similarity between them.

$$ACS_{sim}(i,j) = \frac{\vec{i} \cdot \vec{j}}{\left\|\vec{i}\right\| \cdot \left\|\vec{j}\right\|} \cdot abs_{ib} \tag{4.31}$$

$\vec{i} \cdot \vec{j}$ is the dot-product of the vectors $\vec{i}$ and $\vec{j}$. $\left\|\vec{i}\right\|$ is the magnitude of vector $\vec{i}$ and $\left\|\vec{j}\right\|$ is the magnitude of vector $\vec{j}$. abs_{ib} is defined by Equation 4.12 and Equation 4.13.

$$ACS_{sim}(u,v) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \cdot \|\vec{v}\|} \cdot abs_{ub} \tag{4.32}$$

$\vec{u} \cdot \vec{v}$ is the dot-product of the vectors $\vec{u}$ and $\vec{v}$. $\|\vec{u}\|$ is the magnitude of the vector $\vec{u}$ and $\|\vec{v}\|$ is the magnitude of the vector $\vec{v}$. abs_{ub} is defined by Equation 4.15 and Equation 4.16.

4.3.1.8 Absolute Cosine Co-Rated Similarity

In contrast to the *Absolute Cosine Similarity*, the *Absolute Cosine Co-Rated Similarity* also takes the co-rated objects into account. The results are in the range [0;1], where 0 represents no similarity between the two vectors and 1 full similarity between them. Equation 4.33 is used if the item-based approach is used and Equation 4.34 calculates similarities between users.

$$ACCS_{sim}(i,j) = \frac{\sum_{\mathrm{u \in U}}(R_i \cdot R_j)}{\sqrt{\sum_{\mathrm{u \in U}}(R_i)^2} \cdot \sqrt{\sum_{\mathrm{u \in U}}(R_j)^2}} \cdot abs_{ib} \tag{4.33}$$

$ACCS_{sim}(i,j)$ is the similarity between item i and item j. R_i is the rating of the item i and R_j is the rating of item j. $_{\mathrm{u \in U}}$ is the summation of the users who rated both items i and j. abs_{ib} is defined by Equation 4.12 and Equation 4.13.

$$ACCS_{sim}(u,v) = \frac{\sum_{\mathrm{i \in I}}(R_u \cdot R_v)}{\sqrt{\sum_{\mathrm{i \in I}}(R_u)^2} \cdot \sqrt{\sum_{\mathrm{i \in I}}(R_v)^2}} \cdot abs_{ub} \tag{4.34}$$

$ACCS_{sim}(u,v)$ is the similarity between user u and user v. R_u is the rating of user u and R_v is the rating of user v. $_{\mathrm{i \in I}}$ is the summation of the items rated by both users u and v. abs_{ub} is defined by Equation 4.15 and Equation 4.16.

4.3.1.9 Absolute Cosine Rank Similarity

In contrast to the *Cosine Rank Similarity*, the *Absolute Cosine Rank Similarity* multiplies the result with a factor. This factor is defined by Equation 4.18 or Equation 4.19 if the item-based approach is used. The considering of the user-based approach uses the factor, which is defined by Equation 4.21 or Equation 4.22. The similarity calculation between items is performed by Equation 4.35. Equation 4.36 defines the calculation, if the user-based approach is used. The results of the *Absolute Cosine Rank Similarity* are in the range [0;1], where 0 represents no similarity and 1 full similarity between the two considered vectors.

$$ACRS_{sim}(i,j) = \frac{\vec{i_g} \cdot \vec{j_g}}{\left\|\vec{i_g}\right\| \cdot \left\|\vec{j_g}\right\|} \cdot abs_{rank-ib} \tag{4.35}$$

$\vec{i_g} \cdot \vec{j_g}$ is the dot-product of the vectors $\vec{i_g}$ and $\vec{j_g}$. $\left\|\vec{i_g}\right\|$ is the magnitude of vector $\vec{i_g}$ and $\left\|\vec{j_g}\right\|$ is the magnitude of vector $\vec{j_g}$. $abs_{rank-ib}$ is defined by Equation 4.18 and Equation 4.19.

$$ACRS_{sim}(u,v) = \frac{\vec{u_g} \cdot \vec{v_g}}{\|\vec{u_g}\| \cdot \|\vec{v_g}\|} \cdot abs_{rank-ub} \tag{4.36}$$

$\vec{u_g} \cdot \vec{v_g}$ is the dot-product of the vectors $\vec{u_g}$ and $\vec{v_g}$. $\|\vec{u_g}\|$ is the magnitude of vector $\vec{u_g}$ and $\|\vec{v_g}\|$ is the magnitude of vector $\vec{v_g}$. $abs_{rank-ub}$ is defined by

Equation 4.21 and Equation 4.22.

4.3.1.10 Absolute Cosine Original Rank Similarity

In contrast to the *Absolute Cosine Rank Similarity*, the *Absolute Cosine Original Rank Similarity* takes the factor into account that uses the non-ranked values from the entries of the user-item matrix. The factor is defined by Equation 4.12 or Equation 4.13 if the item-based approach is used. Equation 4.15 or Equation 4.16 defines the factor if the user-based approach is used. The calculation of the *Absolute Cosine Original Rank Similarity* by considering the item-based approach is performed with Equation 4.37. Equation 4.38 defines the calculation by using the user-based approach. The results of the *Absolute Cosine Original Rank Similarity* are in the range [0;1], where 0 represents no similarity and 1 full similarity between the two used vectors.

$$ACORS_{sim}(i,j) = \frac{\vec{i_{og}} \cdot \vec{j_{og}}}{\left\|\vec{i_{og}}\right\| \cdot \left\|\vec{j_{og}}\right\|} \cdot abs_{ib} \tag{4.37}$$

$\vec{i_{og}} \cdot \vec{j_{og}}$ is the dot-product of the vectors $\vec{i_{og}}$ and $\vec{j_{og}}$. $\left\|\vec{i_{og}}\right\|$ is the magnitude of vector $\vec{i_{og}}$ and $\left\|\vec{j_{og}}\right\|$ is the magnitude of vector $\vec{j_{og}}$. abs_{ib} is defined by Equation 4.12 and Equation 4.13.

$$ACORS_{sim}(u,v) = \frac{\vec{u_{og}} \cdot \vec{v_{og}}}{\|\vec{u_{og}}\| \cdot \|\vec{v_{og}}\|} \cdot abs_{ub} \tag{4.38}$$

$\vec{u_{og}} \cdot \vec{v_{og}}$ is the dot-product of the vectors $\vec{u_{og}}$ and $\vec{v_{og}}$. $\|\vec{u_{go}}\|$ is the magnitude of vector $\vec{u_{og}}$ and $\|\vec{v_{og}}\|$ is the magnitude of vector $\vec{v_{og}}$. abs_{ub} is defined by Equation 4.15 and Equation 4.16.

4.3.1.11 Absolute Cosine Rank Co-Rated Similarity

The *Absolute Cosine Rank Co-Rated Similarity* combines the *Cosine Similarity*, the building of ranks, the length of the vectors, and the co-rated approach. The factor is defined by Equation 4.18 and Equation 4.19 if the item-based approach is used. Equation 4.21 and Equation 4.22 defines the factor if the user-based approach is used. The similarity calculation that takes the item-based approach into account is defined by Equation 4.39. Similarities between users are calculated by using Equation 4.40. The results of the *Absolute Cosine Rank Co-Rated Similarity* are in the range [0;1], where 0 represents no similarity and 1 full similarity between the two used vectors.

$$ACRCS_{sim}(i,j) = \frac{\sum_{\mathrm{u}\in\mathrm{U}}(Rg_i \cdot Rg_j)}{\sqrt{\sum_{\mathrm{u}\in\mathrm{U}}(Rg_i)^2} \cdot \sqrt{\sum_{\mathrm{u}\in\mathrm{U}}(Rg_j)^2}} \cdot abs_{rank-ib} \tag{4.39}$$

$ACRCS_{sim}(i,j)$ is the similarity between item i and item j. Rg_i is the ranked rating of item i and Rg_j is the ranked rating of item j. $_{\mathrm{u}\in\mathrm{U}}$ is the summation of the users who rated both items i and j. $abs_{rank-ib}$ is defined by Equation 4.18 and Equation 4.19.

$$ACRCS_{sim}(u,v) = \frac{\sum_{\mathrm{i}\in\mathrm{I}}(Rg_u \cdot Rg_v)}{\sqrt{\sum_{\mathrm{i}\in\mathrm{I}}(Rg_u)^2} \cdot \sqrt{\sum_{\mathrm{i}\in\mathrm{I}}(Rg_v)^2}} \cdot abs_{rank-ub} \tag{4.40}$$

$ACRCS_{sim}(u,v)$ is the similarity between user u and user v. Rg_u is the ranked rating of user u and Rg_v is the ranked rating of user v. $_{\mathrm{i \in I}}$ is the summation of the items rated by both users u and v. $abs_{rank-ub}$ is defined by Equation 4.21 and Equation 4.22.

4.3.1.12 Absolute Cosine Original Rank Co-Rated Similarity

In contrast to the *Absolute Cosine Rank Co-Rated Similarity*, the *Absolute Cosine Original Rank Co-Rated Similarity* uses the non-ranked entries from the user-item matrix for the building of the factor. The factor is defined by Equation 4.12 and Equation 4.13 if the item-based approach is used. Equation 4.15 and Equation 4.16 defines the factor by using the user-based approach. The similarity calculation between items is performed by Equation 4.41. Equation 4.42 defines the calculation by using the user-based approach. The results of the *Absolute Cosine Original Rank Co-Rated Similarity* are in the range [0;1], where 0 represents no similarity and 1 full similarity between the two used vectors.

$$ACORCS_{sim}(i,j) = \frac{\sum_{\text{u}\in\text{U}}(Rg_i \cdot Rg_j)}{\sqrt{\sum_{\text{u}\in\text{U}}(Rg_i)^2} \cdot \sqrt{\sum_{\text{u}\in\text{U}}(Rg_j)^2}} \cdot abs_{ib} \tag{4.41}$$

$ACORCS_{sim}(i,j)$ is the similarity between item i and item j. Rg_i is the ranked rating of item i and Rg_j is the ranked rating of item j. ${}_{\text{u}\in\text{U}}$ is the summation of the users who rated both items i and j. abs_{ib} is defined by Equation 4.12 and Equation 4.13.

$$ACORCS_{sim}(u,v) = \frac{\sum_{\text{i}\in\text{I}}(Rg_u \cdot Rg_v)}{\sqrt{\sum_{\text{i}\in\text{I}}(Rg_u)^2} \cdot \sqrt{\sum_{\text{i}\in\text{I}}(Rg_v)^2}} \cdot abs_{ub} \tag{4.42}$$

$ACORCS_{sim}(u, v)$ is the similarity between user u and user v. Rg_u is the ranked rating of user u and Rg_v is the ranked rating of user v. $_{\mathrm{i}\in\mathrm{I}}$ is the summation of the items rated by both users u and v. abs_{ub} is defined by Equation 4.15 and Equation 4.16.

4.3.1.13 Adjusted Cosine Rank Similarity

In contrast to the *Adjusted Cosine Similarity*, the *Adjusted Cosine Rank Similarity* introduces the using of the ranks. Equation 4.43 defines the calculation by considering the item-based approach. Equation 4.44 takes the user-based approach into account. The results are in the range [-1;+1], where -1 represents full negative similarity, 1 full positive similarity, and 0, no similarity between the two considered objects.

$$AJCRS_{sim}(i,j) = \frac{\sum_{\mathrm{u \in U}}(Rg_{u,i} - \overline{Rg}_u)(Rg_{u,j} - \overline{Rg}_u)}{\sqrt{\sum_{\mathrm{u \in U}}(Rg_{u,i} - \overline{Rg}_u)^2}\sqrt{\sum_{\mathrm{u \in U}}(Rg_{u,j} - \overline{Rg}_u)^2}} \tag{4.43}$$

$AJCRS_{sim}(i,j)$ is the similarity between item i and item j. $\overline{Rg}_u$ is the ratings' average from user u by using the ranks of the user-item matrix. $u \in U$ is the summation of the users who rated item i and item j. $Rg_{u,i}$ is the ranks' rating of item i from user u and $Rg_{u,j}$ is the ranks' rating of item j from user u.

$$AJCRS_{sim}(u,v) = \frac{\sum_{\mathrm{i \in I}}(Rg_{u,i} - \overline{Rg}_i)(R_{v,i} - \overline{Rg}_i)}{\sqrt{\sum_{\mathrm{i \in I}}(Rg_{u,i} - \overline{Rg}_i)^2}\sqrt{\sum_{\mathrm{i \in I}}(Rg_{v,i} - \overline{Rg}_i)^2}} \tag{4.44}$$

$AJCRS_{sim}(u,v)$ is the similarity between user u and user v. $\overline{Rg}_i$ is the ratings' average from item i by using the ranks of the user-item matrix. $i \in I$ is the summation of the items that are rated by user u and user v. $Rg_{u,i}$ is the ranks' rating of item i from user u and $R_{v,i}$ is the ranks' rating of item i from user v.

4.3.1.14 Overview

Table 4.3.1.14 presents an overview of the algorithms and clarifies which newly developed algorithm is based on which idea of SotA algorithms. Additionally the techniques, such as the abs-factor, the building of ranks, or the usage of the co-rated approach, are listed as well. The appreciations of the algorithms are defined by Table 5.1.

Newly Developed	**SotA**	**Techniques**
AC	PC	Abs-Factor
ARC	SRC	Abs-Factor, Rank
AORC	PC,SRC	Abs-Factor, Rank
CCS	CS,PC	Co-Rated
CRS	CS,SRC	Rank
CRCS	CS,PC,SRC	Rank, Co-Rated
ACS	CS	Abs-Factor
ACCS	CS,PC	Abs-Factor, Co-Rated
ACRS	CS,SRC	Abs-Factor, Rank
ACORS	CS,PC,SRC	Abs-Factor, Rank
ACRCS	CS,SRC	Abs-Factor, Rank, Co-Rated
ACORCS	CS,SRC	Abs-Factor, Rank, Co-Rated
AJCRS	AJCS,SRC	Rank

Table 4.2: Overview of the newly developed algorithms and the basis of these algorithms with the used techniques

4.3.2 K-Nearest Neighbours

The calculated similarities which are delivered by the using the presented collaborative-filtering algorithms are used to find the k-nearest neighbours. The k-nearest neighbour approach is described in Section 2.2.2.2 in more detail. The proposed system calculates the similarities between the active user or item and the other users/items within the given user-item matrix. Within the next step, the system orders the achieved similarities in decreasing order. The system will use this information for the building of the new user-item matrix that contains just the active user/item and its k-nearest neighbours.

4.3.3 Prediction Calculations

In order to prove that an algorithm delivers the most accurate results, the proposed system of this thesis calculates predictions. The prediction calculation is defined by Equation 2.14. This equation is also known as *Weighted Sum*.

The prediction calculation is accomplished with every item within the given user-item matrix. Therefore each entry must be deleted from the user-item matrix. The calculated predictions are needed to exploit the error rates, which is described in Section 4.3.4.

4.3.3.1 Prediction Truncations

The results of the prediction calculation can deliver values that are not within the rating range. The rating range of the proposed system is [0;5]. If the calculated prediction is above the maximum value or below it, the proposed system truncates the predictions as follows. If the prediction is <0, the system will set it to 0. If the prediction is >5, the system sets it to 5. The usefulness of the prediction truncation is proved by an evaluation. This evaluation compares the error rates of the system by using and not using the truncation of predictions. The results of this evaluation are presented in Section 5.1.

4.3.4 Error Calculations

The proposed system shall deliver the most accurate algorithm. Therefore an error exploitation is needed. Since most of the researchers of the related work use the MAE or the RMSE, the proposed system also uses these error rates. The RMSE is based on the MSE. The equations are described in Section 2.2.2.2.

The comparison between the results from the related work and the proposed system shall prove the usefulness of the dynamic multi-algorithm collaborative-filtering system.

4.3.5 Dynamic Selection of Most Accurate Algorithm

The dynamic selection of the most accurate collaborative-filtering algorithm takes all the above mentioned approaches into account. Figure 4.6 illustrates the process of the proposed system. The output of this process is the most accurate filtering algorithm. In order to take recent preferences into account, the process could be started e.g., if a user logs into a recommendation system or if a user changed the own preferences.

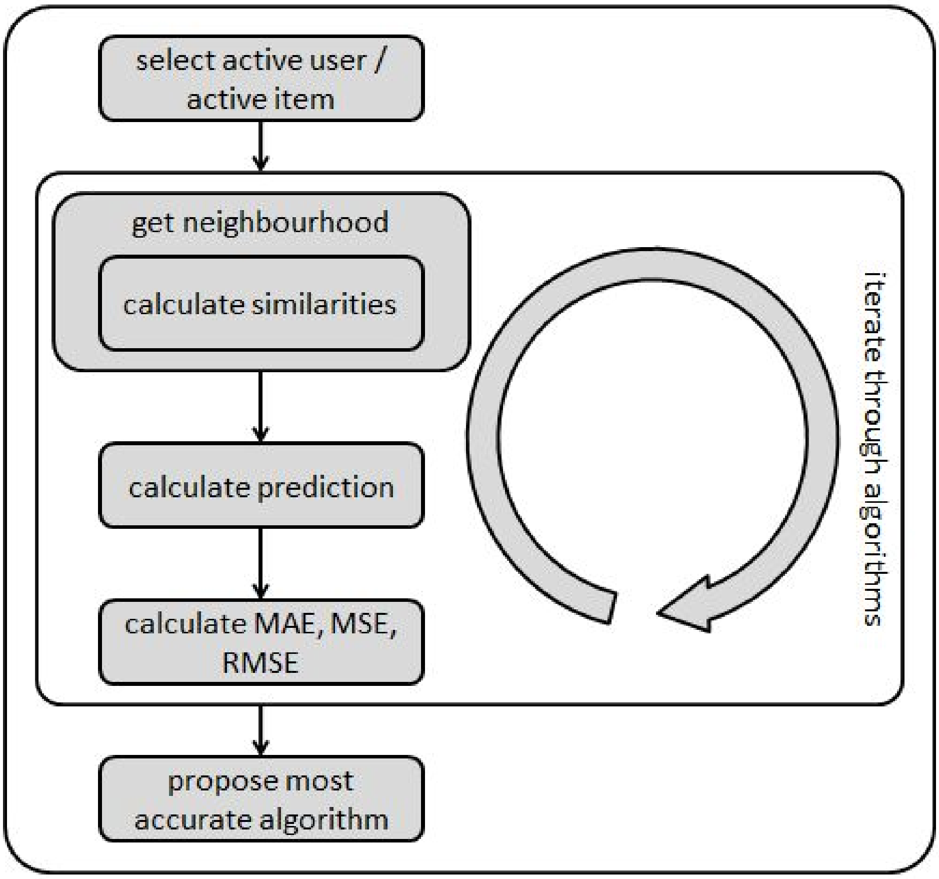

select active user / active item
get neighbourhood
calculate similarities
calculate prediction
calculate MAE, MSE, RMSE
propose most accurate algorithm
iterate through algorithms

Figure 4.6: The procedure of the proposed dynamic multi-algorithm collaborative-filtering system

Select active user/item

The first step selects the active user if similar users shall be found. If similar items shall be found, the system uses the active item.

Get neighbourhood

The neighbourhood is represented by the k-nearest neighbours. They will be achieved by the calculation of the similarities between the active user/item and the other users/items within the used user-item matrix. The system delivers a new user-item matrix, which only includes the k-nearest neighbours and the active user/item.

Calculate predictions

The user-item matrix with the k-nearest neighbours will be used to calculate predictions. This step predicts every entry of the active user/item by considering the similarities from each k-nearest neighbour. If a prediction is not within the rating scale, the system truncates the prediction as described above.

Calculate MAE, MSE, and RMSE

The calculated predictions are used to calculate the errors. The system compares the original value of each entry from the active user/item with the prediction of these entries. This comparison is needed to calculate the error rates, such as the MAE, MSE, and the RMSE.

Iterate through algorithms

The above mentioned steps will be realized with every algorithm which is included in the proposed dynamic multi-algorithm collaborative-filtering system.

Propose the most accurate filtering algorithm

The last step proposes the most accurate collaborative-filtering algorithm for the currently active user/item.

4.4 Summary

This chapter presented the developed and researched recommendation system. It introduced the reader to user profiling and described the creation of these profiles in an explicit and implicit manner. Besides the creation of the user profiles, the filtering techniques were described. Since the main objective of this thesis tackles collaborative-filtering techniques, this chapter described the content-based filtering only briefly. The main part of this chapter presented the collaborative-filtering algorithms. It presented the newly developed algorithms which overcome researched weaknesses of SotA algorithms. In addition to the filtering algorithms, the dynamic approach, which is able to reduce the error rates significantly compared to existing approaches, was also described.

Chapter 5

Evaluation

The evaluation of this thesis considers different user-matrices.

The experiments use the dataset which is the result of a survey that was undertaken at the THM. The results of this survey are presented in Table 3.1. Members and students took part in this survey. They were asked to rate genres that are specified by an ETSI Standard for Service Information. This standard specified twelve main genres, which are presented in Table 3.2. Each respondent could rate these genres by setting stars within a range [0;5]. 0 stars represent no interest in the selected genre and 5 stars represent definite interest in the selected genre. The output of this survey is a quite small user-item matrix and could represent a community, like a family or a block of flats. In order to take huge communities into account, the evaluation of this thesis also considers a dataset from MovieLens. This dataset contains ratings from 943 users and 1682 movies (items).

The evaluation section is organized as follows: Section 5.1 presents the results of the prediction truncation. It compares the error rates that uses the prediction truncation with the results that do not use the truncation of predictions. Section 5.2 presents the results of the error rates by using the presented algorithms. The experiments of this section do not use the k-nearest neighbour approach or the proposed dynamic approach. Section 5.3 presents the results which have been achieved by using the k-nearest neighbour approach. Section 5.4 presents the results that use the proposed dynamic multi-algorithm collaborative-filtering system. Section 5.5 compares the error rates from existing systems with the error rate of the proposed system. Section 5.6 concludes the evaluation chapter and the achieved results.

Since the names of the algorithms are quite long, Table 5.1 presents the abbreviations of the algorithms that are used in this chapter.

Abbreviation	**Algorithm**
PC	Pearson-r Correlation
AC	Absolute Correlation
SRC	Spearman Rank Correlation
ARC	Absolute Rank Correlation
AORC	Absolute Original Rank Correlation
CS	Cosine Similarity
CCS	Cosine Co-Rated Similarity
CRS	Cosine Rank Similarity
CRCS	Cosine Rank Co-Rated Similarity
ACS	Absolute Cosine Similarity
ACCS	Absolute Cosine Co-Rated Similarity
ACRS	Absolute Cosine Rank Similarity
ACORS	Absolute Cosine Original Rank Similarity
ACRCS	Absolute Cosine Rank Co-Rated Similarity
ACORCS	Absolute Cosine Original Rank Co-Rated Similarity
AJCS	Adjusted Cosine Similarity
AJCRS	Adjusted Cosine Rank Similarity

Table 5.1: The abbreviations of the considered collaborative-filtering algorithms

5.1 Prediction Truncation

This section presents the comparison of the results that have been achieved by considering the prediction truncation. Figure 5.1 and Figure 5.2 present the results. The MAEs have been achieved by predicting every entry within the MovieLens dataset. The results prove that the prediction truncation is able to decrease the error rate for each algorithm.

Figure 5.1 presents the MAE by considering the item-based approach. It illustrates the calculated MAEs that have been achieved without the truncation of the prediction and the MAEs that have been achieved by using the prediction truncation. The results prove that the predication truncation decreases the MAE.

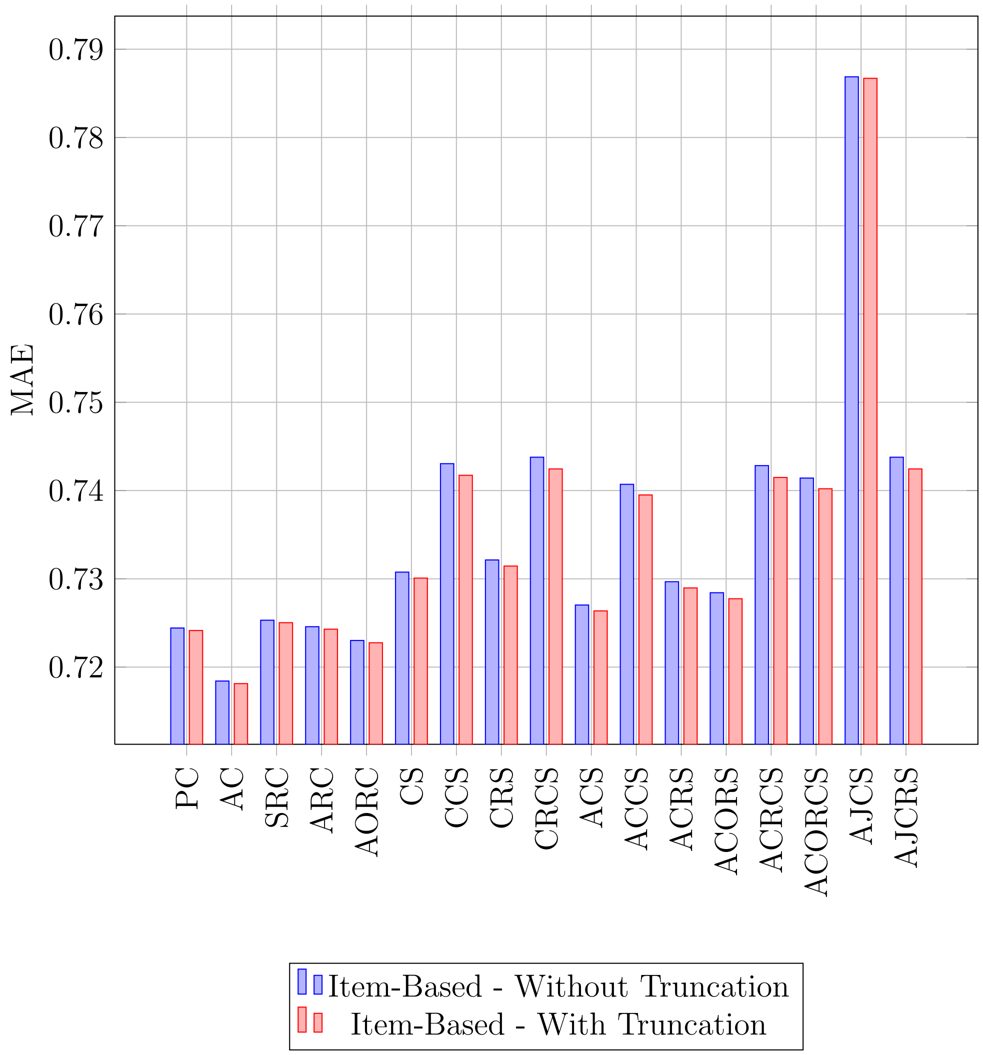


Figure 5.1: Comparison between MAE item-based without truncation and MAE item-based with truncation

The results of the MAEs by considering the user-based approach are presented in Figure 5.2. The figure presents the calculated MAEs that have been achieved without the truncation of the prediction and the MAEs that have been achieved by using the prediction truncation. The results prove that the predication truncation decreases the MAE.

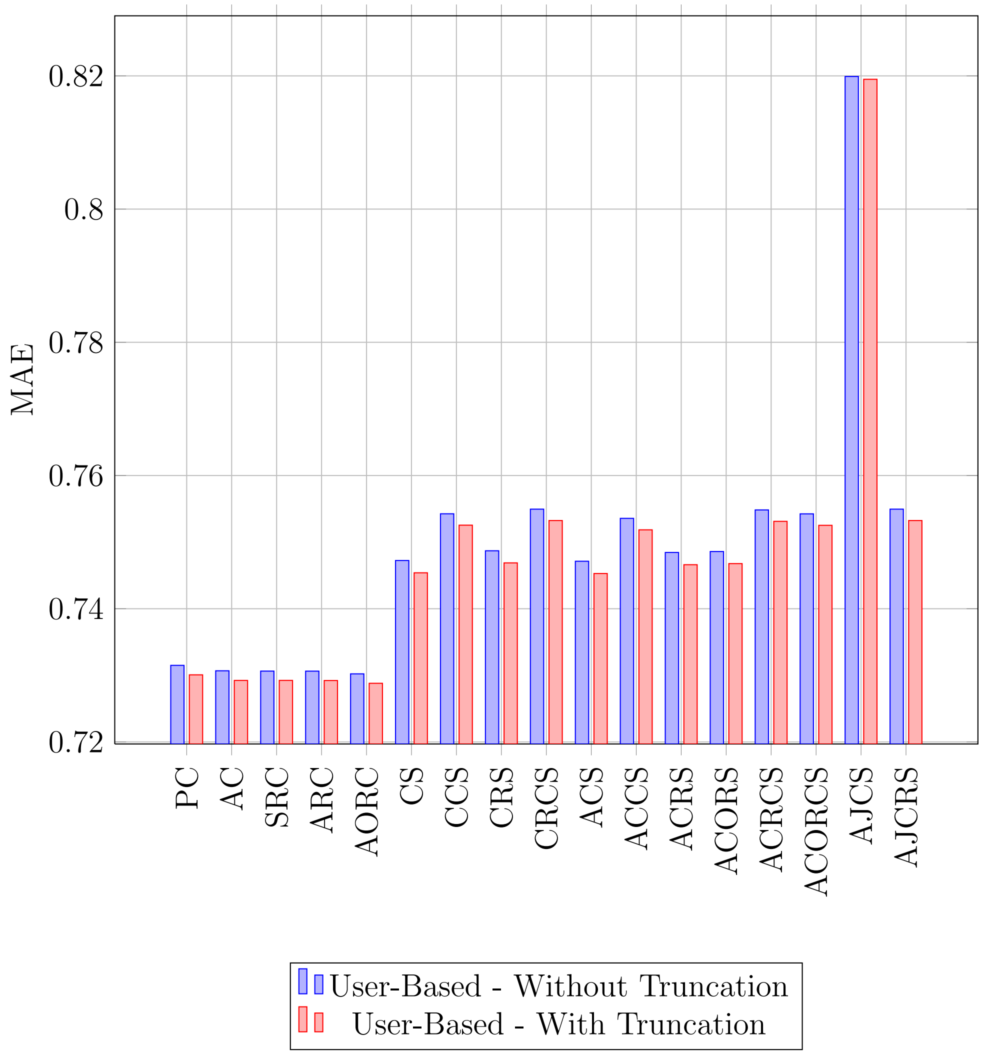


Figure 5.2: Comparison between MAE user-based without truncation and MAE user-based with truncation

5.1.1 Conclusion

Section 5.1 compares the error rates that have been achieved by taking the truncation of the predictions into account with the results that do not truncate the predictions. Figure 5.1 presents the MAE by considering the item-based approach and Figure 5.2 presents the results by taking the user-based approach into account. The experiment uses the dataset from MovieLens. The results prove that the predications' truncation is able to decrease the error rates.

5.2 Without Neighbourhood

5.2.1 Survey

This section presents the results of accomplished tests by using the data from the survey which are presented by Table 3.1.

5.2.1.1 Item-Based

Figures 5.3-5.7 present the results revealing the MAE of different tests. These results prove that every test delivers another algorithm which is the most adequate for calculating the predictions by considering the item-based approach.

The *Absolute Cosine Similarity* produces a MAE of 1.2680 by considering all ten users, which is shown in Figure 5.3.

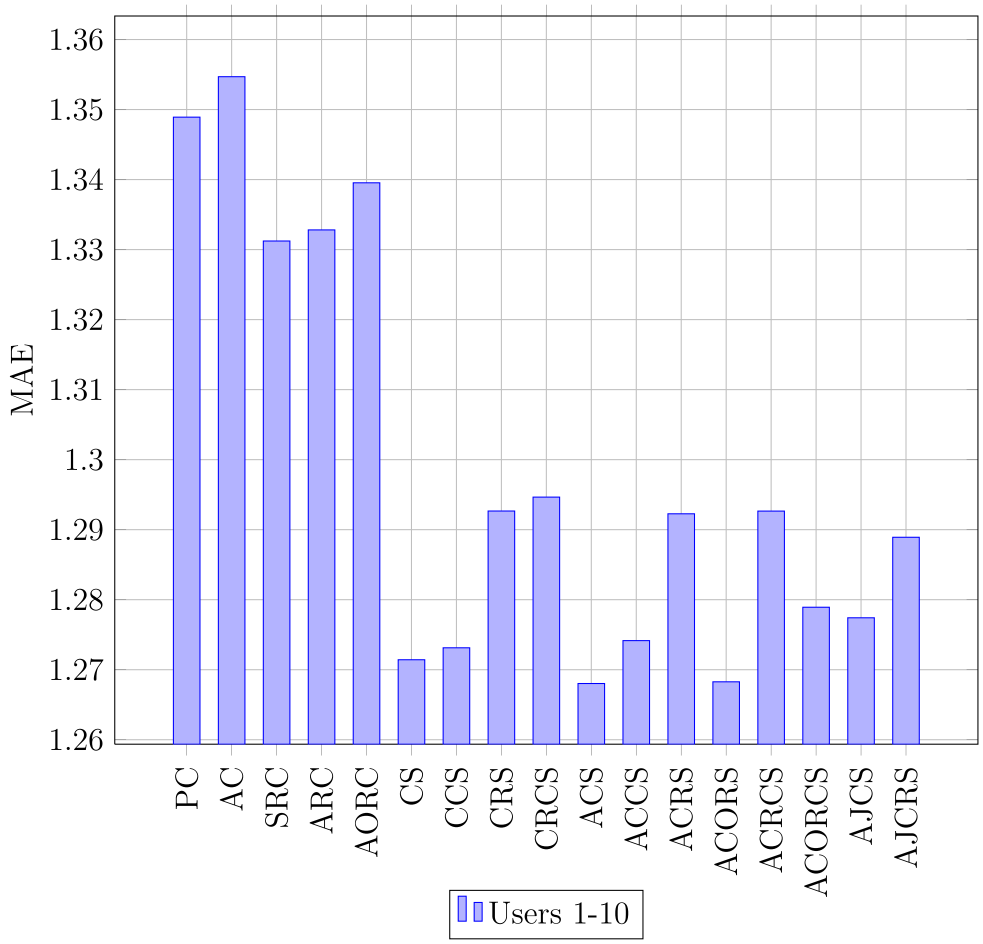


Figure 5.3: MAEs by considering users 1-10 by the usage of the item-based approach and taking the dataset from the survey into account

Figure 5.4 illustrates that the most adequate algorithm by considering user 1 - user 5 is the *Adjusted Cosine Rank Similarity*. This algorithms produces a MAE of 1.3380.

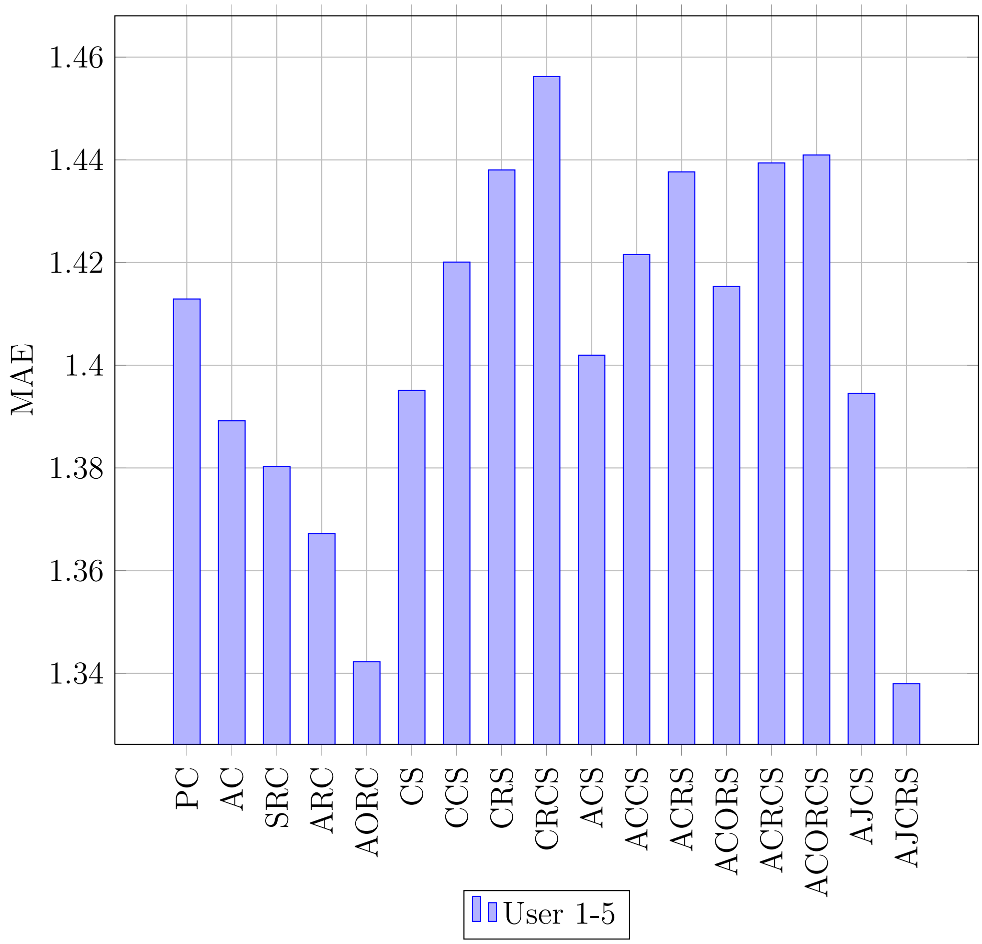


Figure 5.4: MAEs by considering users 1-5 by the usage of the item-based approach and taking the dataset from the survey into account

The *Absolute Cosine Similarity* is the most adequate algorithm by using user 6 - user 10. This is visualized in Figure 5.5. The calculated MAE of this algorithm is 1.1737.

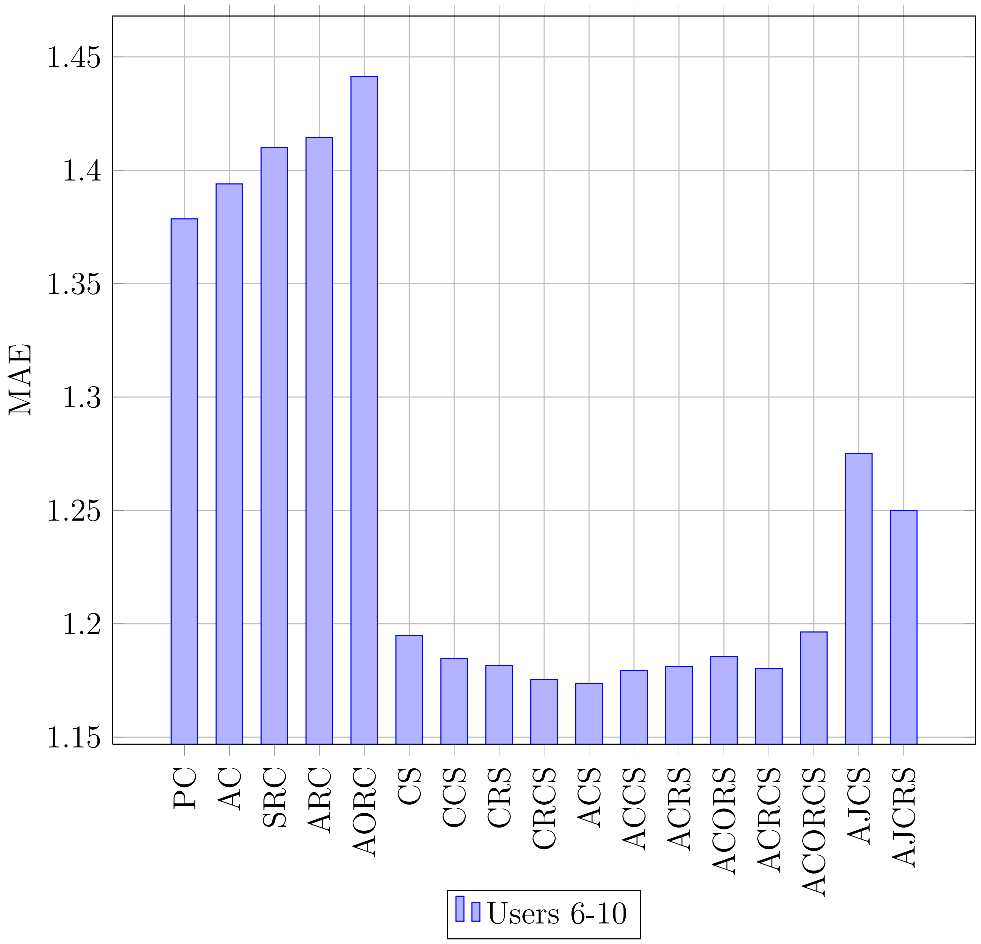


Figure 5.5: MAEs by considering users 6-10 by the usage of the item-based approach and taking the dataset from the survey into account

Figure 5.6 shows that the algorithm with the MAE of 1.4421 by considering user 1, user 3, user 5, user 7, and user 9 is the *Absolute Cosine Rank Similarity*.

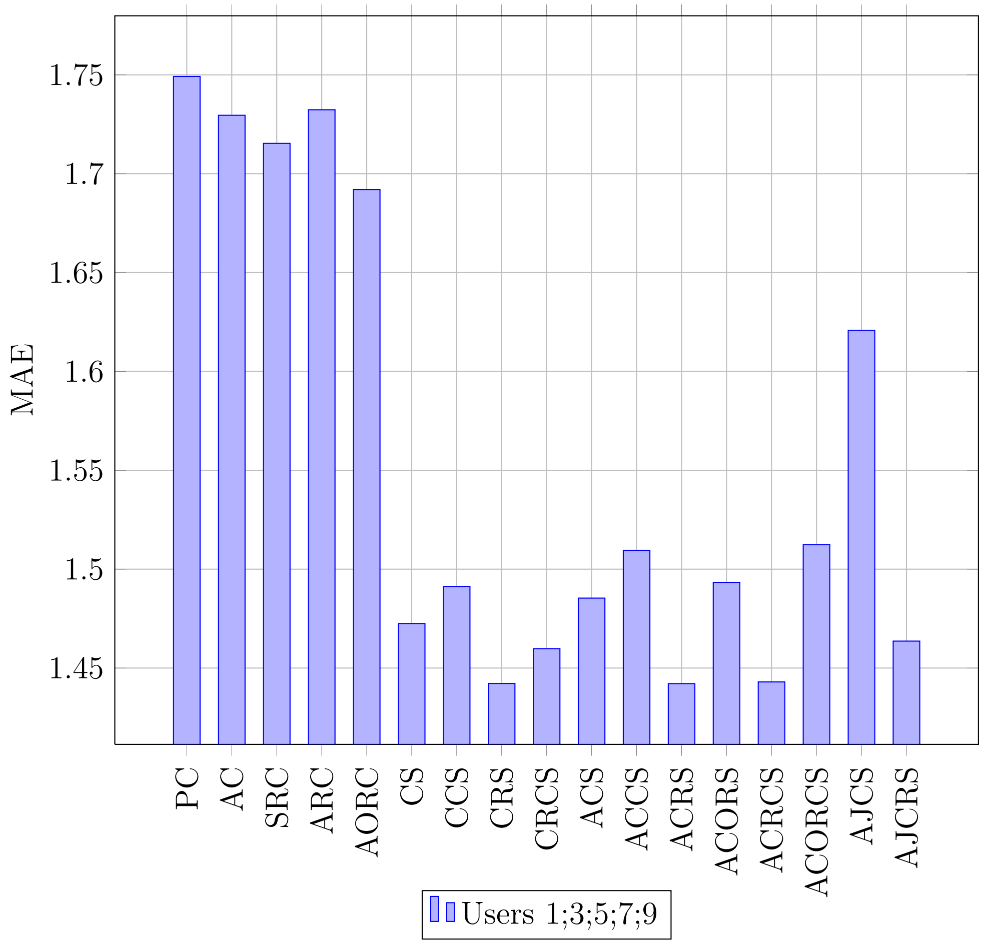


Figure 5.6: MAEs by considering users 1,3,5,7,9 by the usage of the item-based approach and taking the dataset from the survey into account

The *Absolute Original Rank Correlation* is the most adequate algorithm by using user 2, user 4, user 6, user 8, and user 10 and produces a MAE of 1.1192, which is presented in Figure 5.7.

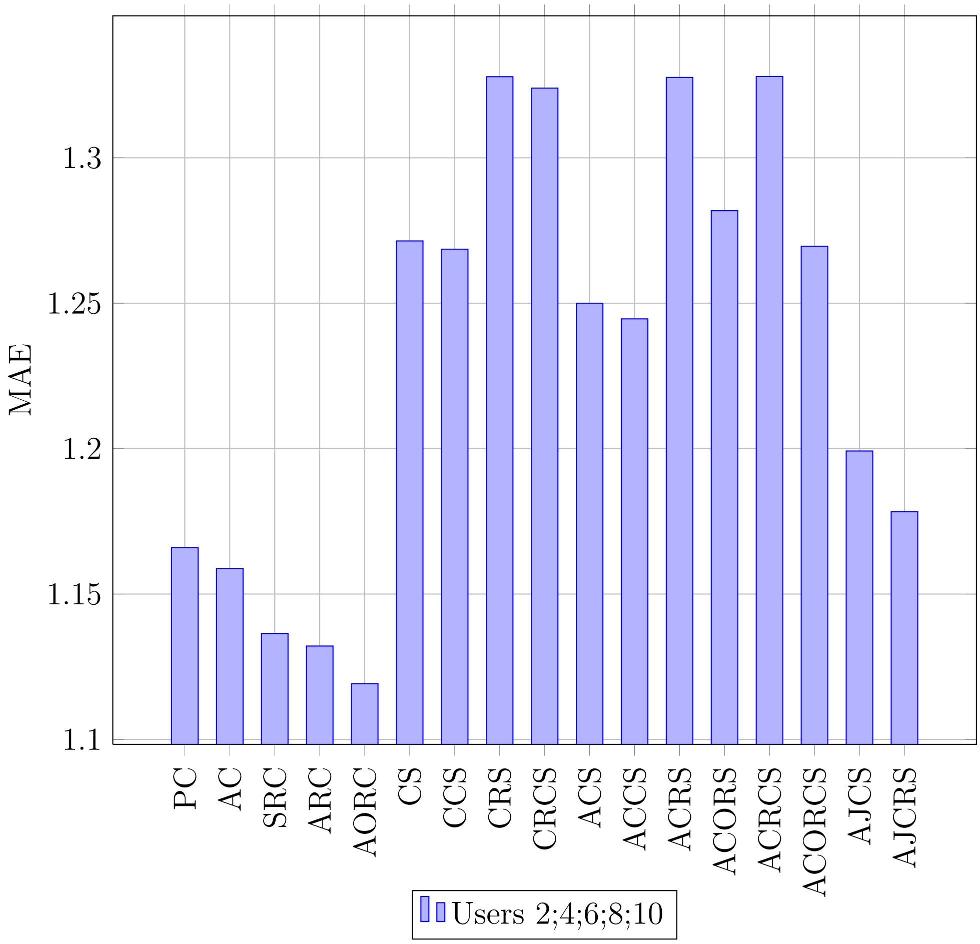


Figure 5.7: MAEs by considering users 2,4,6,8,10 by the usage of the item-based approach and taking the dataset from the survey into account

Figure 5.8 shows the results at a glance and shall visualize the fluctuation of the MAEs more clearly.

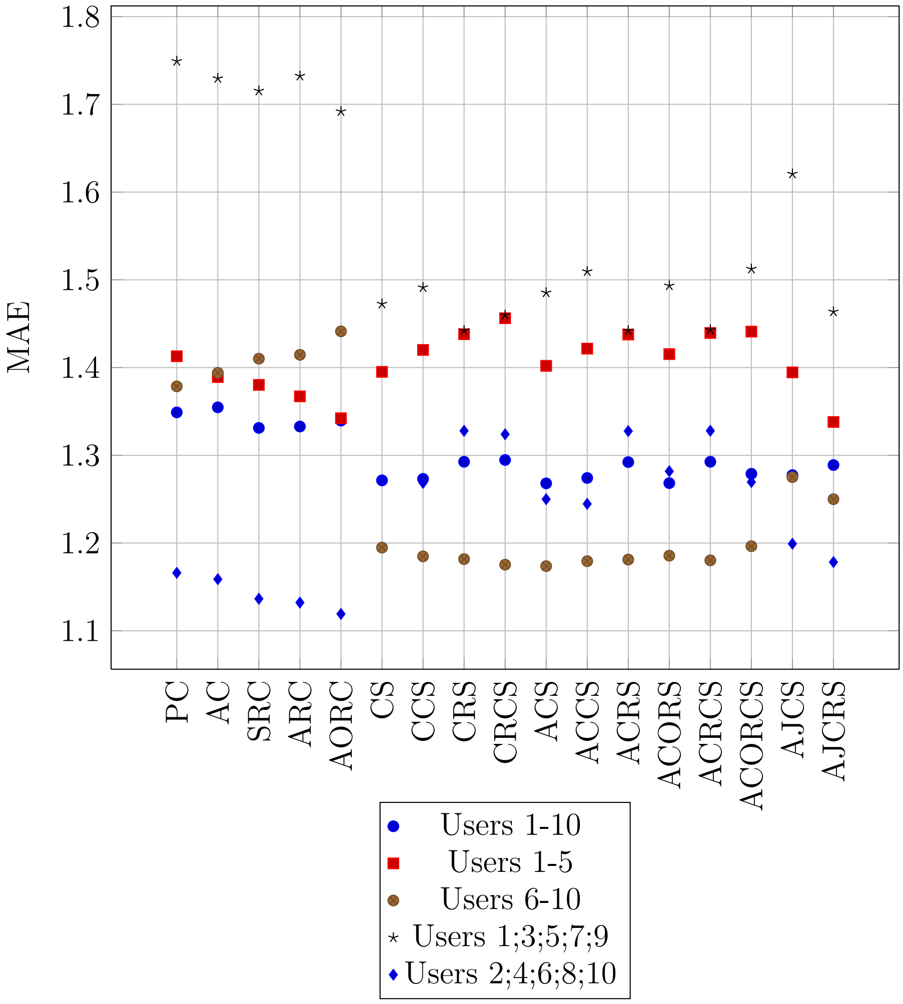


Figure 5.8: MAEs of the different used user-item matrices at a glance

5.2.1.2 User-Based

Figure 5.9-5.13 present the results revealing the MAE of different tests by considering the user-based approach.

Figures 5.9 shows that the *Pearson-r Correlation* produces a MAE of 1.2748 by considering all ten users.

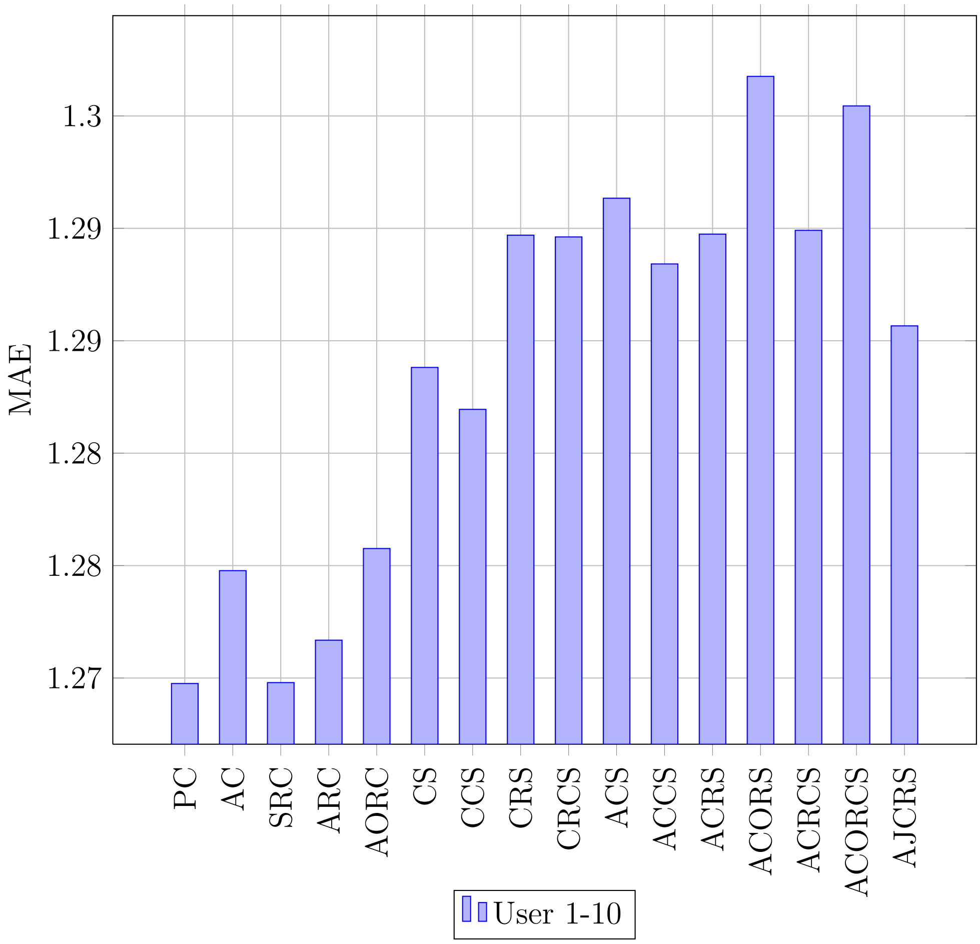


Figure 5.9: MAEs by considering users 1-10 by the usage of the user-based approach and taking the dataset from the survey into account

The most adequate algorithm by considering user 1 - user 5 is the *Cosine Similarity*. This algorithm produces a MAE of 1.4291, which is visualized in Figure 5.10.

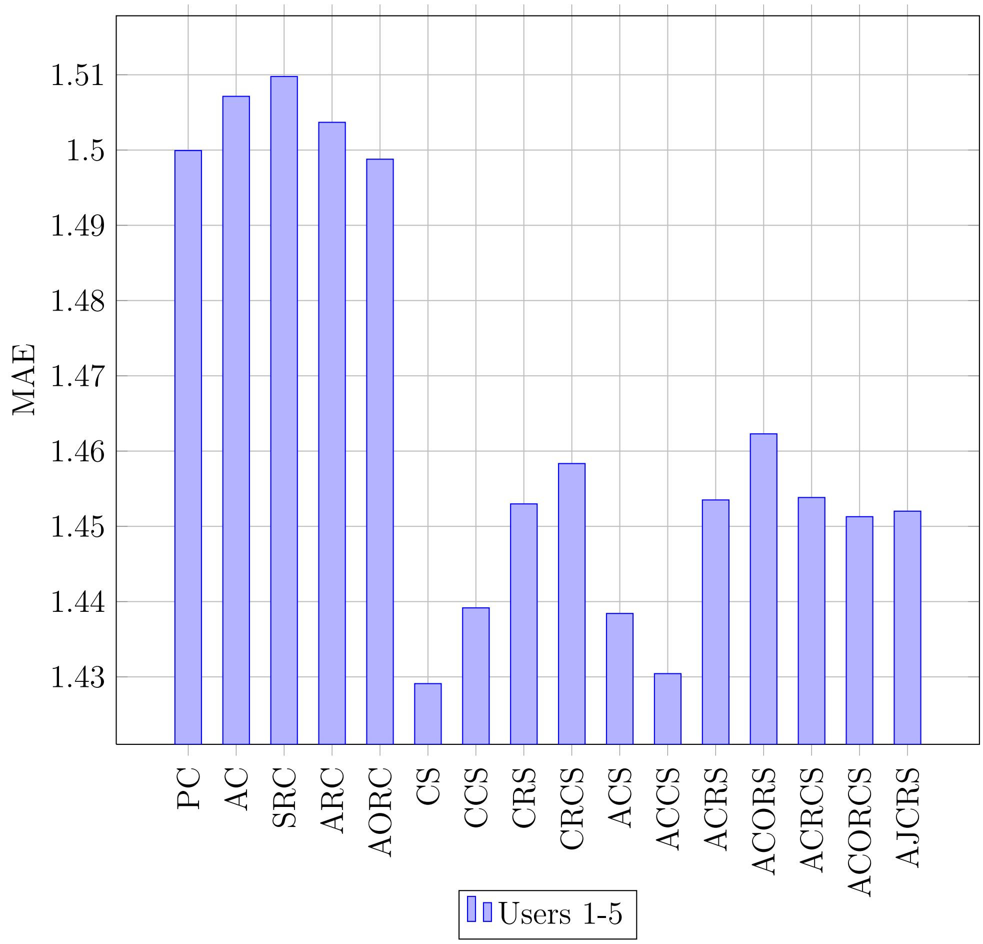


Figure 5.10: MAEs by considering users 1-5 by the usage of the user-based approach and taking the dataset from the survey into account

Figure 5.11 shows that the *Cosine Co-Rated Similarity* is the most adequate algorithm by using user 6 - user 10. This algorithm produces a MAE 1.1775.

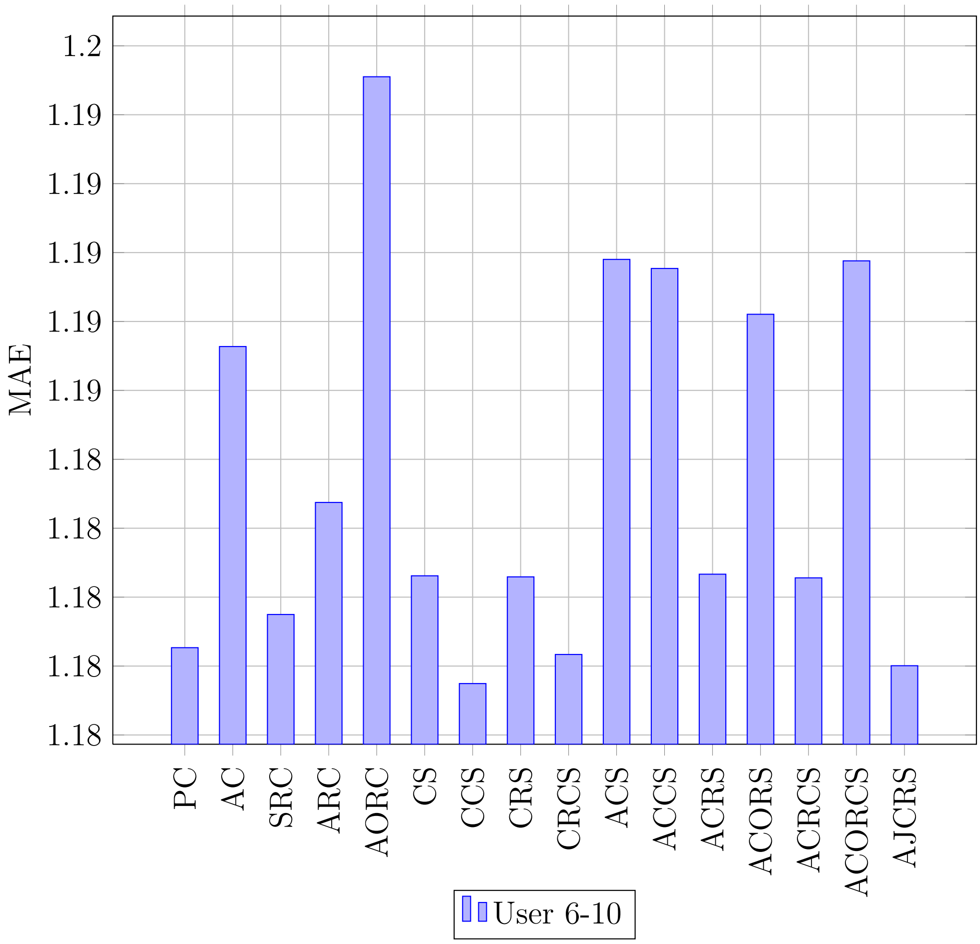


Figure 5.11: MAEs by considering users 6-10 by the usage of the user-based approach and taking the dataset from the survey into account

The algorithm with the lowest MAE of 1.4527, which considers user 1, user 3, user 5, user 7 and user 9, is the *Cosine Co-Rated Similarity*. This is visualized in Figure 5.6.

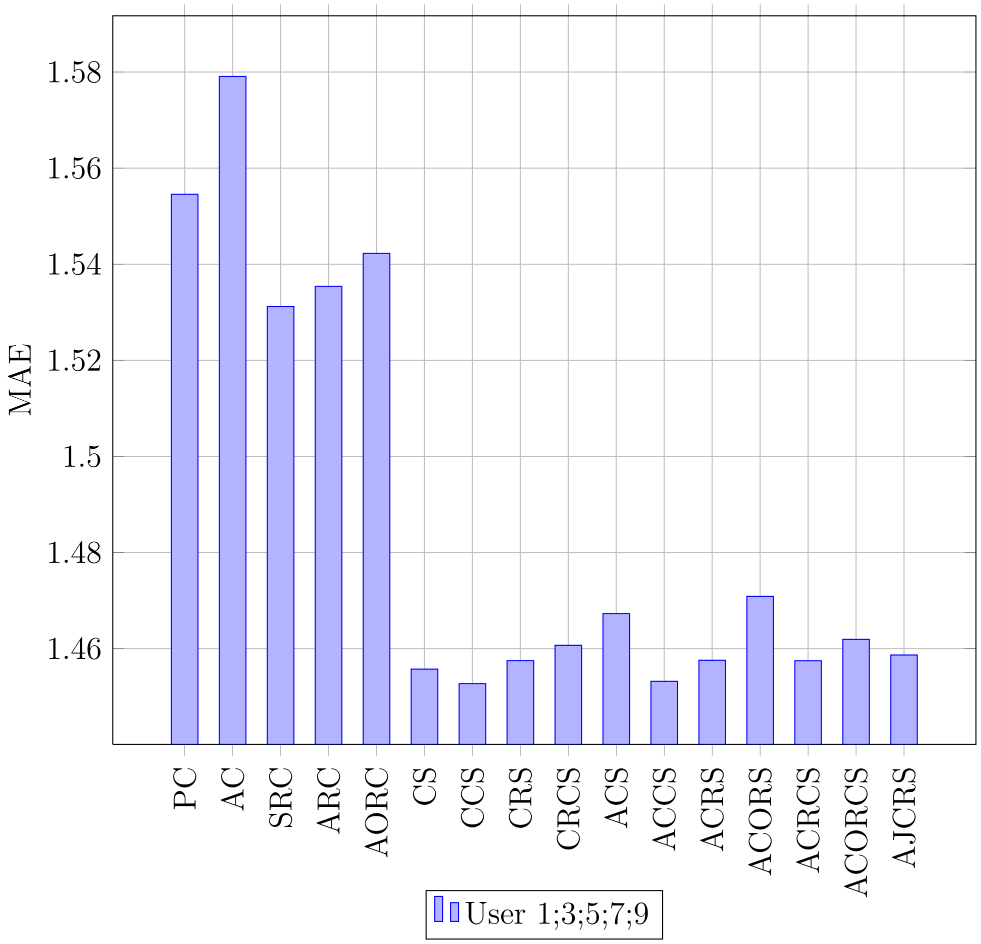


Figure 5.12: MAEs by considering users 1,3,5,7,9 by the usage of the user-based approach and taking the dataset from the survey into account

The *Pearson-r Correlation* is the most adequate one and this test uses user 2, user 4, user 6, user 8, and user 10. Figure 5.13 shows that this algorithm produces a MAE of 1.2654.

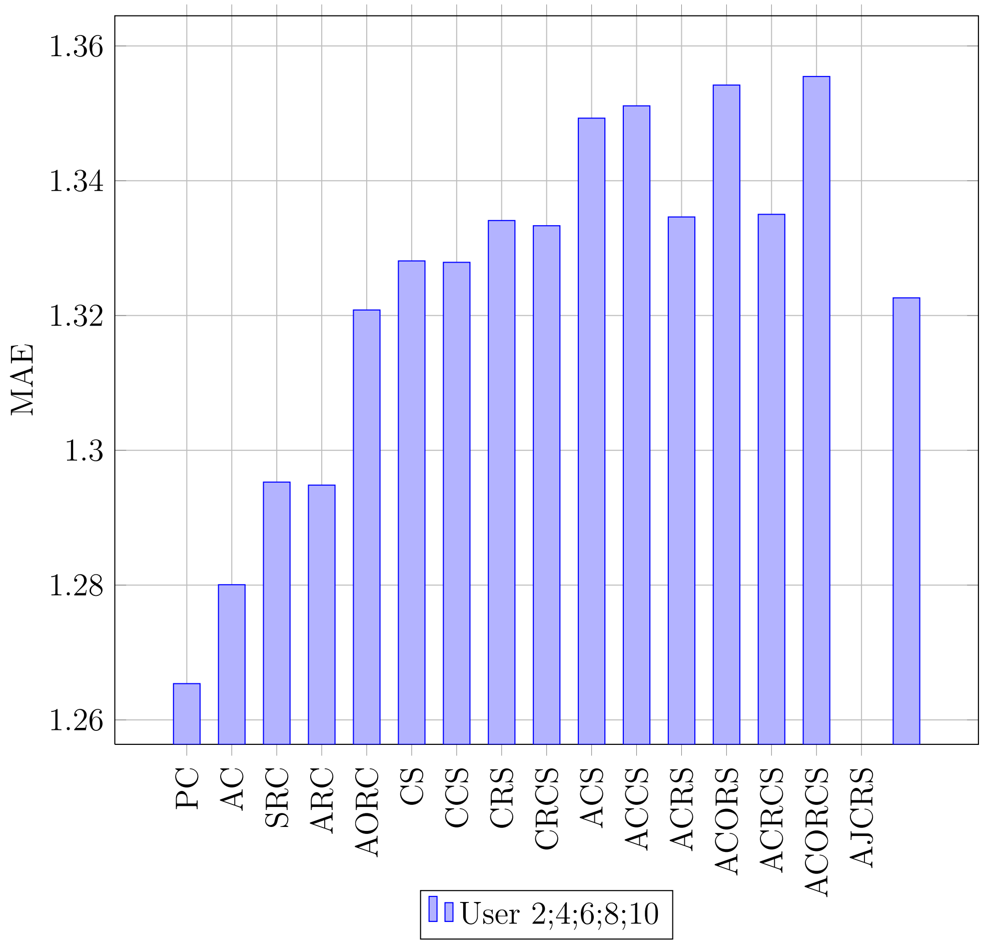


Figure 5.13: MAEs by considering users 2,4,6,8,10 by the usage of the user-based approach and taking the dataset from the survey into account

Figure 5.14 presents the results of the user-based approach at a glance.

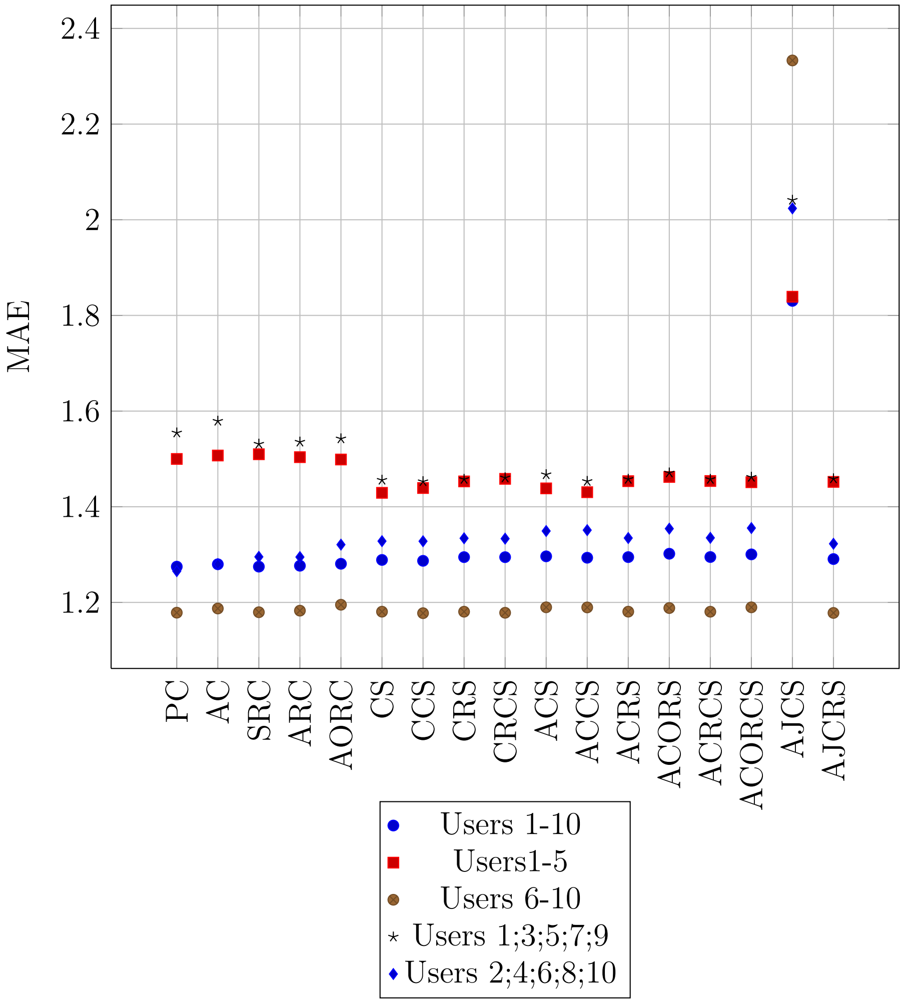


Figure 5.14: MAEs of the different used user-item matrices at a glance

Figure 5.15 shows the results without the *Adjusted Cosine Similarity*, because the MAE of this algorithm is significantly higher than the other MAEs.

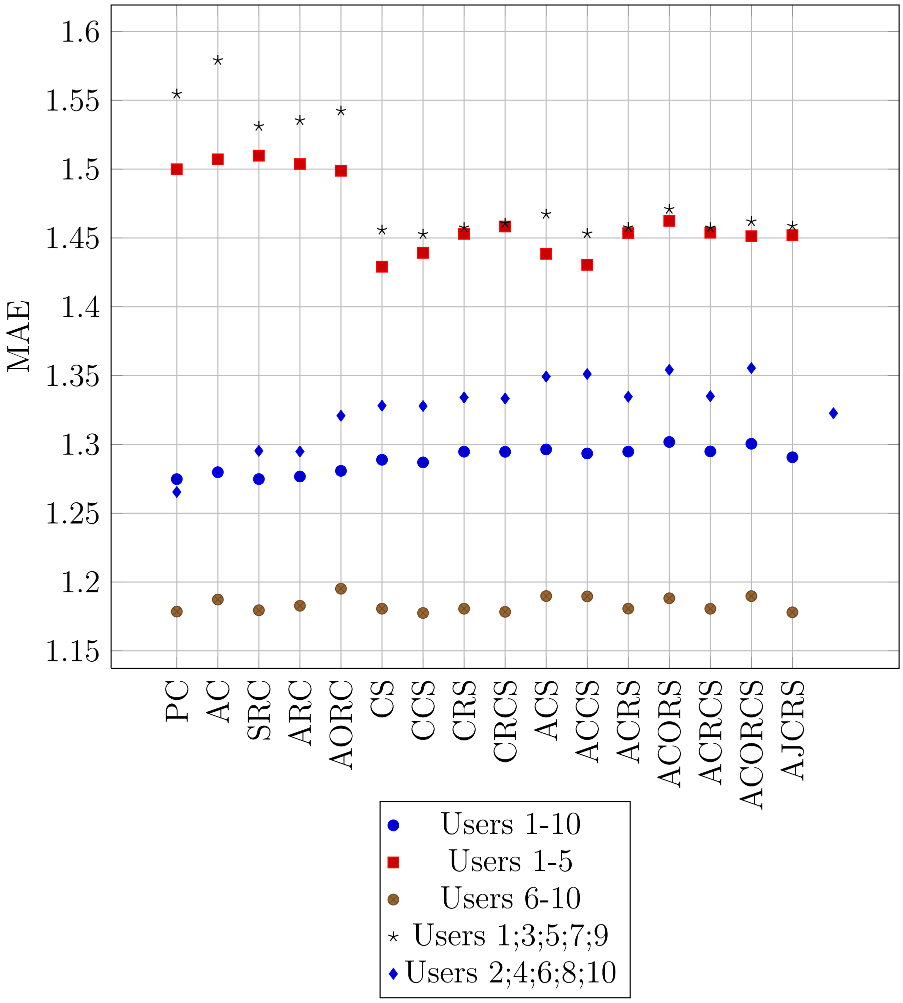


Figure 5.15: MAEs of the different used user-item matrices at a glance without the Adjusted Cosine Similarity

5.2.2 MovieLens

The dynamic multi-algorithm collaborative-filtering system has been developed for a recommendation system that can consider large and small datasets. However, this section shall show that this system is not limited to a quite small user group. It can also be used in a huge community. Therefore these tests use data from MovieLens.

MovieLens is a recommendation system, which is able to recommend movies. This recommendation system is able to recommend movies by using the user profiles. In addition it is able to find similar contents to a currently selected movie. Besides these features, MovieLens also offers the opportunity to use the user-item matrices which were built by users. These matrices are used to test the developed multi-algorithm collaborative-filtering system and shall show the usefulness of the novel developed algorithms as well. The file which includes the user-item matrix contains 943 users and 1682 movies. The following tests use the user-item matrix from MovieLens.

Figure 5.16 presents the MAE by the usage of the item-based approach. The results show that the *Absolute Correlation* produces the lowest MAE by considering the item-based approach. The MAE is 0.718127535.

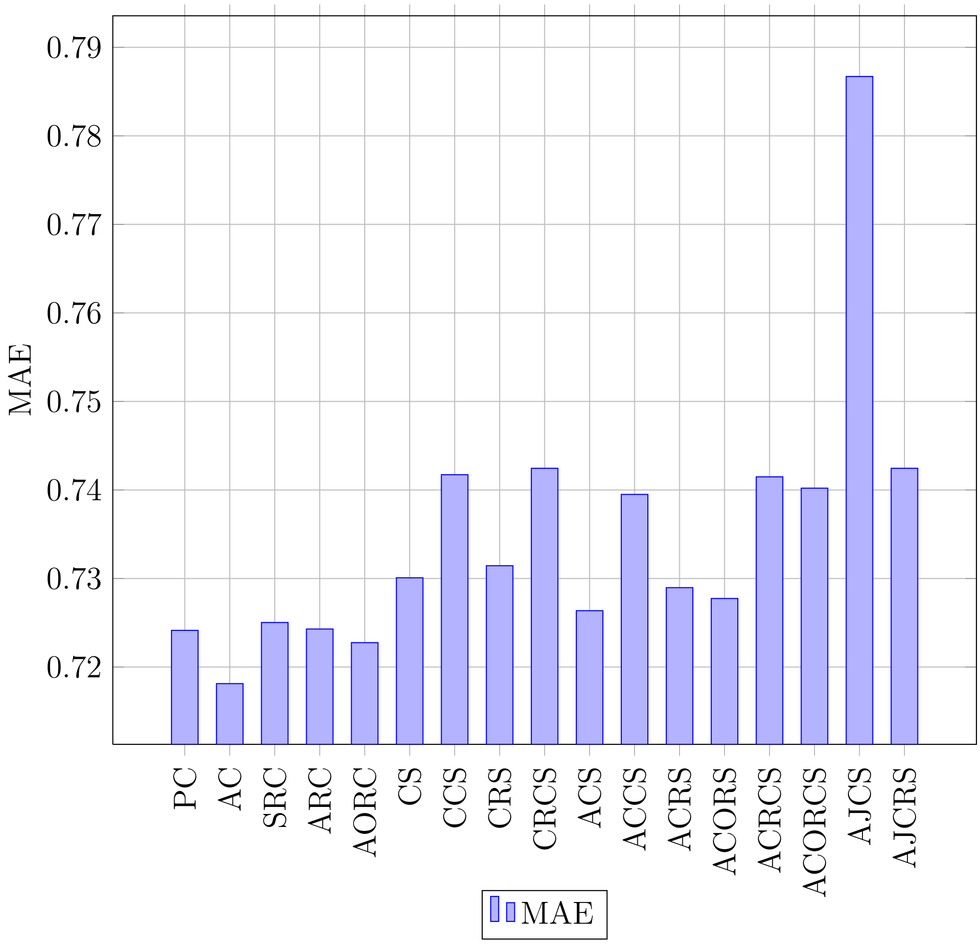


Figure 5.16: MAE by considering the item-based approach and by using the dataset from MovieLens

Figure 5.17 presents the MSE by the usage of the item-based approach. The results show that the *Absolute Correlation* produces the lowest MSE by considering the item-based approach. The MSE is 0.830160531.

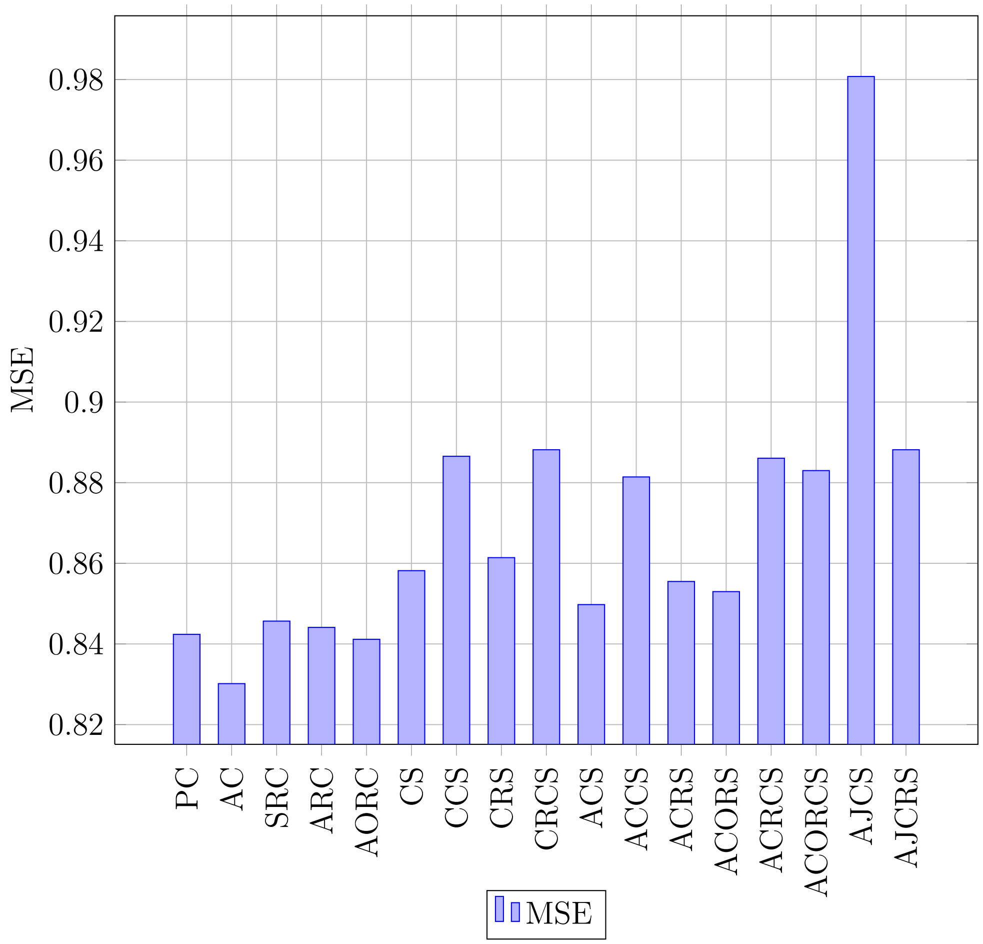


Figure 5.17: MSE by considering the item-based approach and by using the dataset from MovieLens

Figure 5.18 presents the RMSE by the usage of the item-based approach. The results show that the *Absolute Correlation* produces the lowest RMSE by considering the item-based approach. The RMSE is 0.911131457.

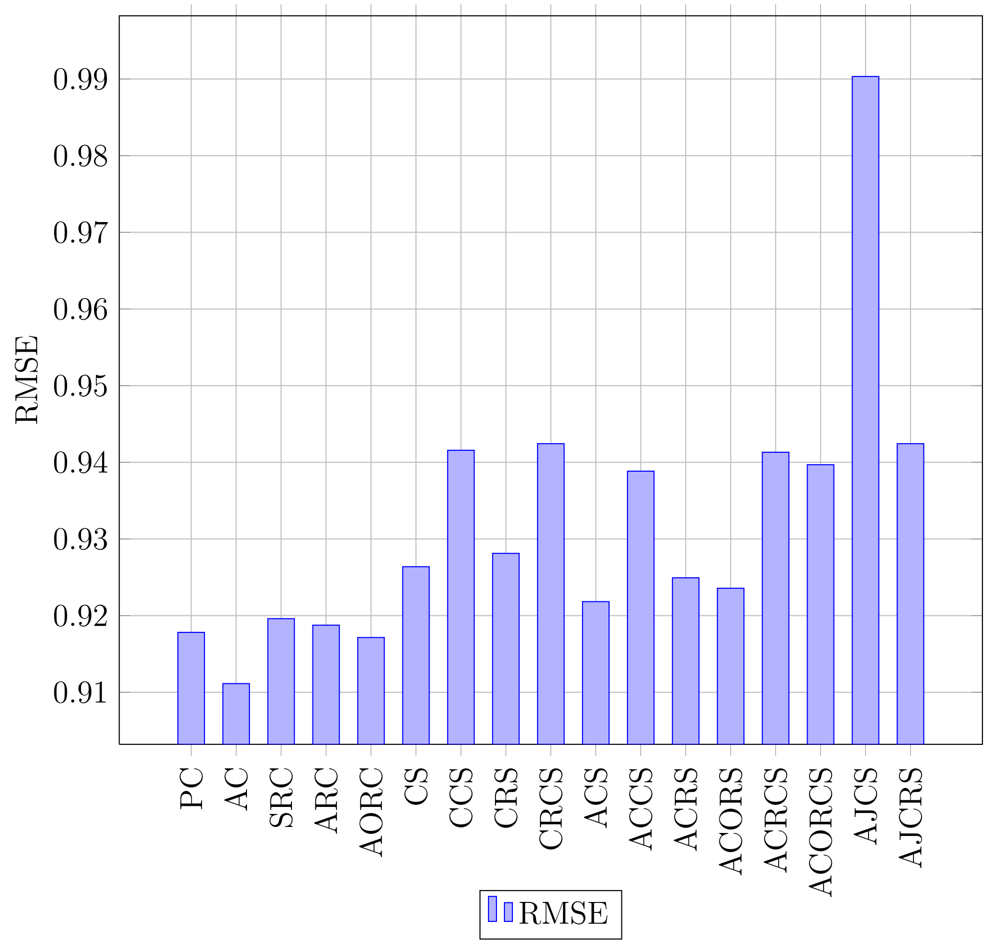


Figure 5.18: RMSE by considering the item-based approach and by using the dataset from MovieLens

Figure 5.19 presents the MAE by the usage of the user-based approach. The results show that the *Absolute Original Rank Correlation* produces the lowest MAE by considering the user-based approach. The MAE is 0.728804945.

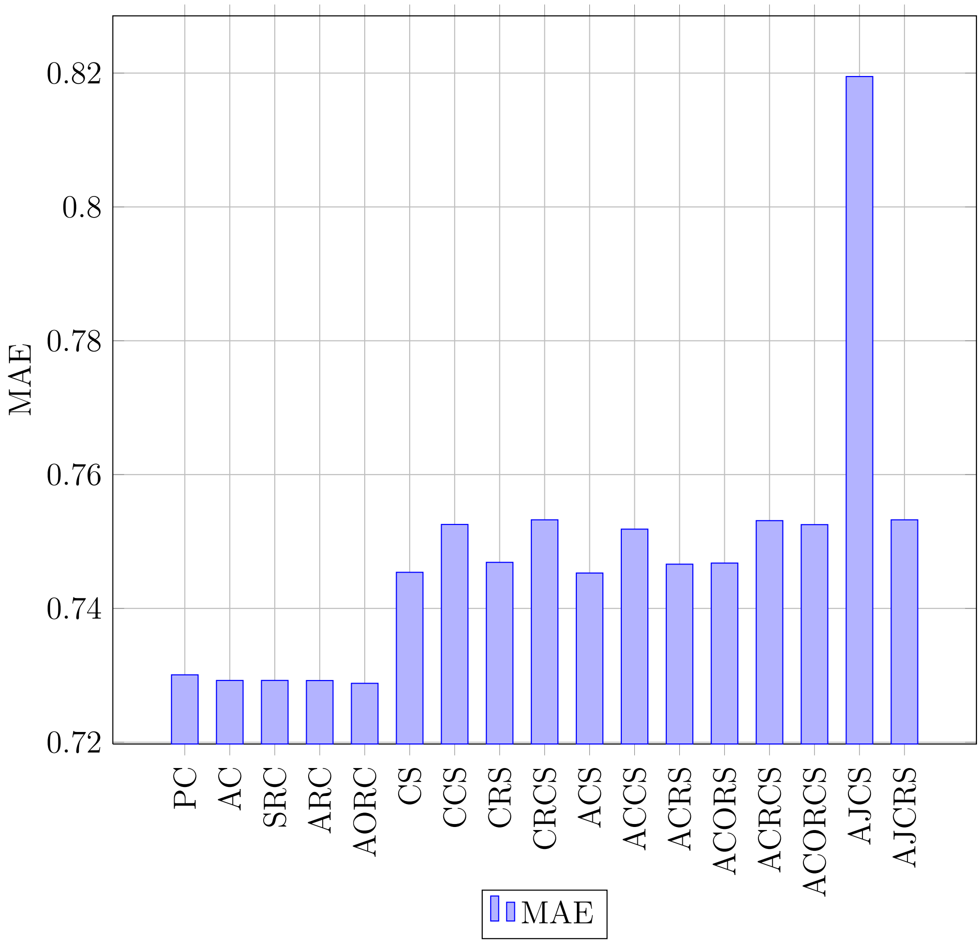


Figure 5.19: MAE by considering the user-based approach and by using the dataset from MovieLens

Figure 5.20 presents the MSE by the usage of the user-based approach. The results show that the *Absolute Original Rank Correlation* produces the lowest MSE by considering the user-based approach. The MSE is 0.858242675.

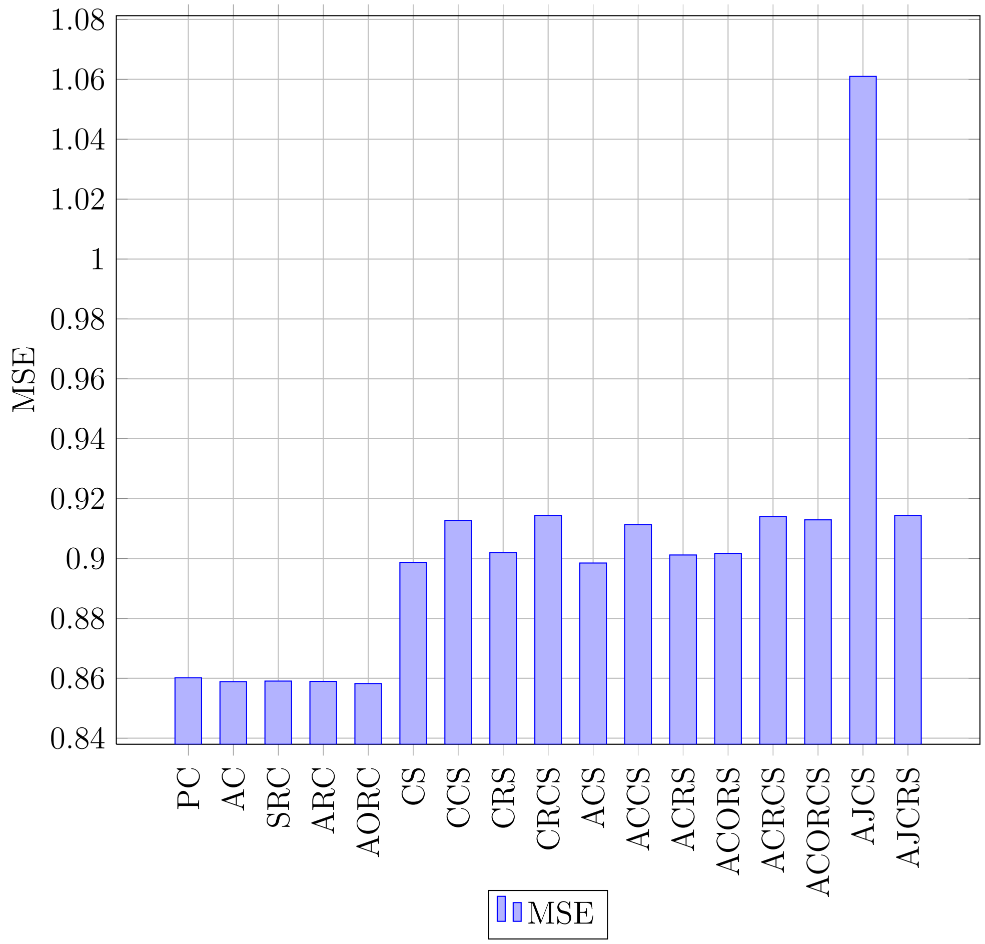


Figure 5.20: MSE by considering the user-based approach and by using the dataset from MovieLens

Figure 5.21 presents the RMSE by the usage of the user-based approach. The results show that the *Absolute Original Rank Correlation* produces the lowest RMSE by considering the user-based approach. The RMSE is 0.926413879.

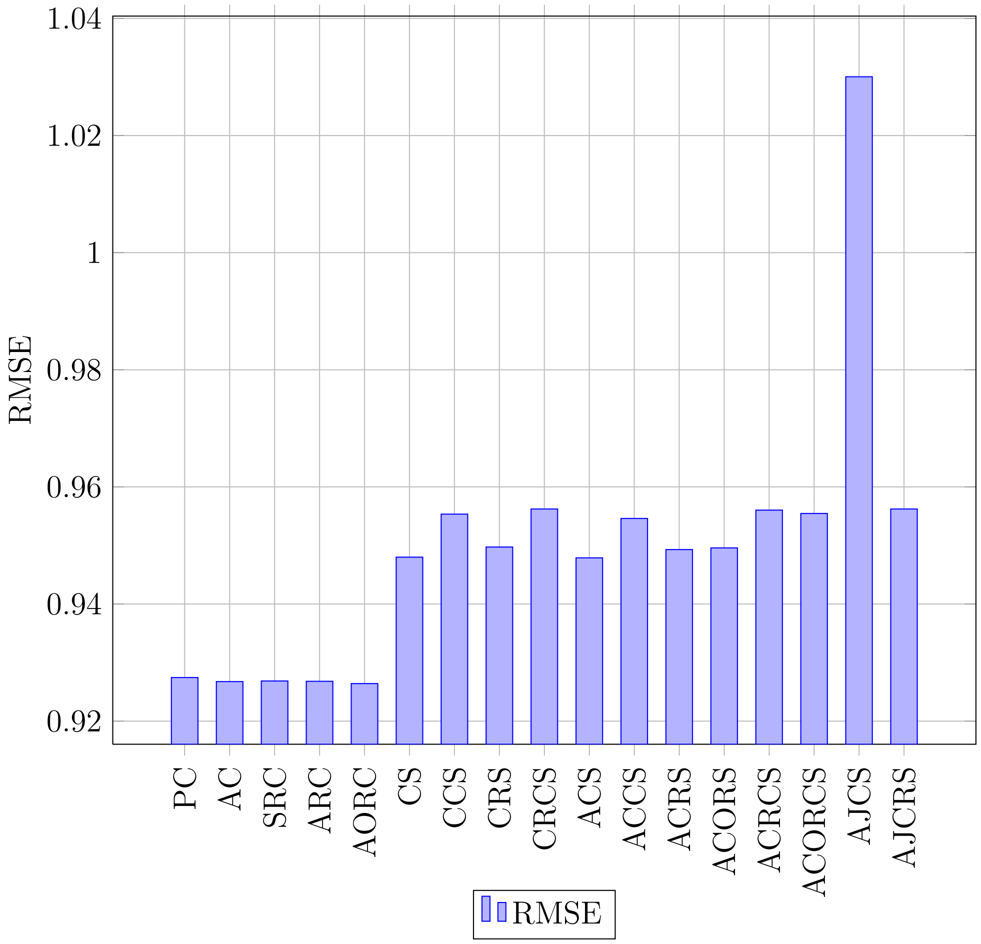


Figure 5.21: RMSE by considering the user-based approach and by using the dataset from MovieLens

5.2.2.1 Performance - MovieLens

The calculations which use the user-item matrix from MovieLens were performed by a server. This server includes several Central Processing Unit (CPU)s. Each CPU has a power of 2.66 Ghz and each Personal Computer (PC) contains 4 GByte Random Access Memory (RAM). Figure 5.23 shows the results of the performance by considering the user-based approach and Figure 5.22 presents the results by using the item-based approach. Each algorithm uses the user-item matrix from MovieLens, which contains ratings from 943 users and 1682 items. Each item from each user was used for the testing. This means that each entry within the user-item matrix has been predicted by the test.

Figure 5.22 presents the duration of the calculation by using the item-based approach.

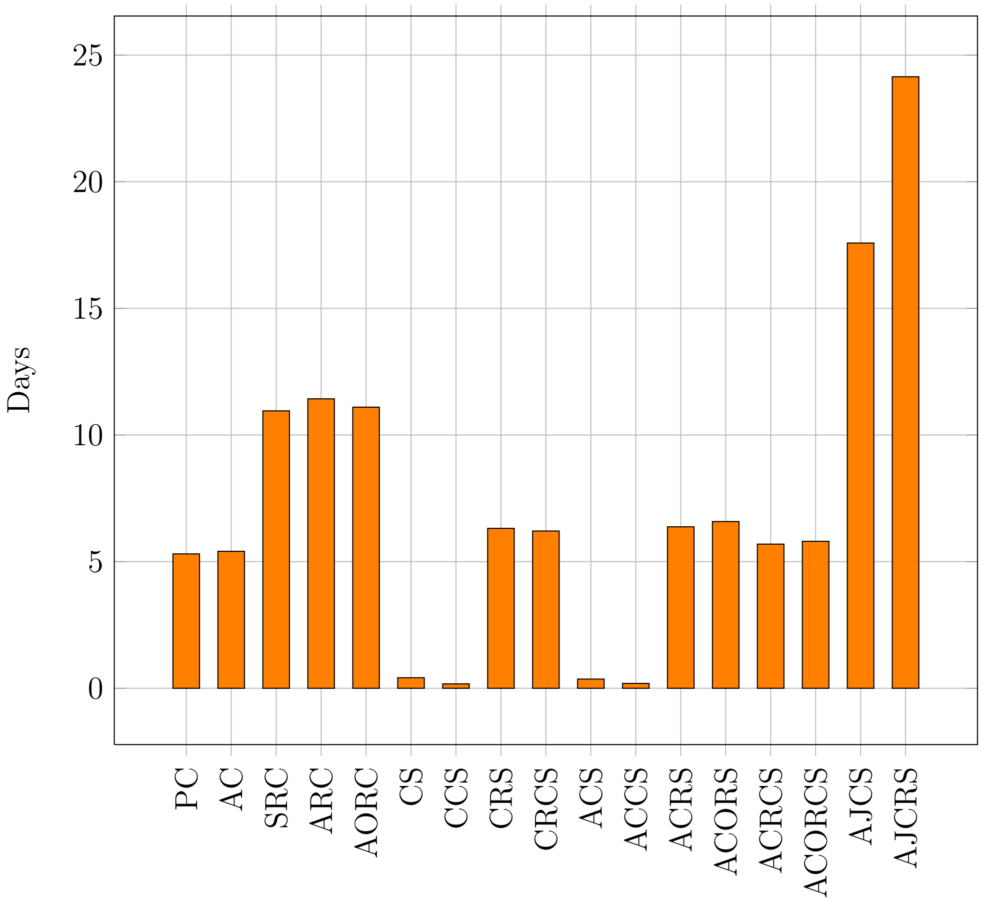


Figure 5.22: Calculation durations of each algorithm by taking the item-based approach into account and by the usage of the dataset from MovieLens

Figure 5.23 presents the calculation durations by taking the user-based approach into account. The experiments do not use an active user.

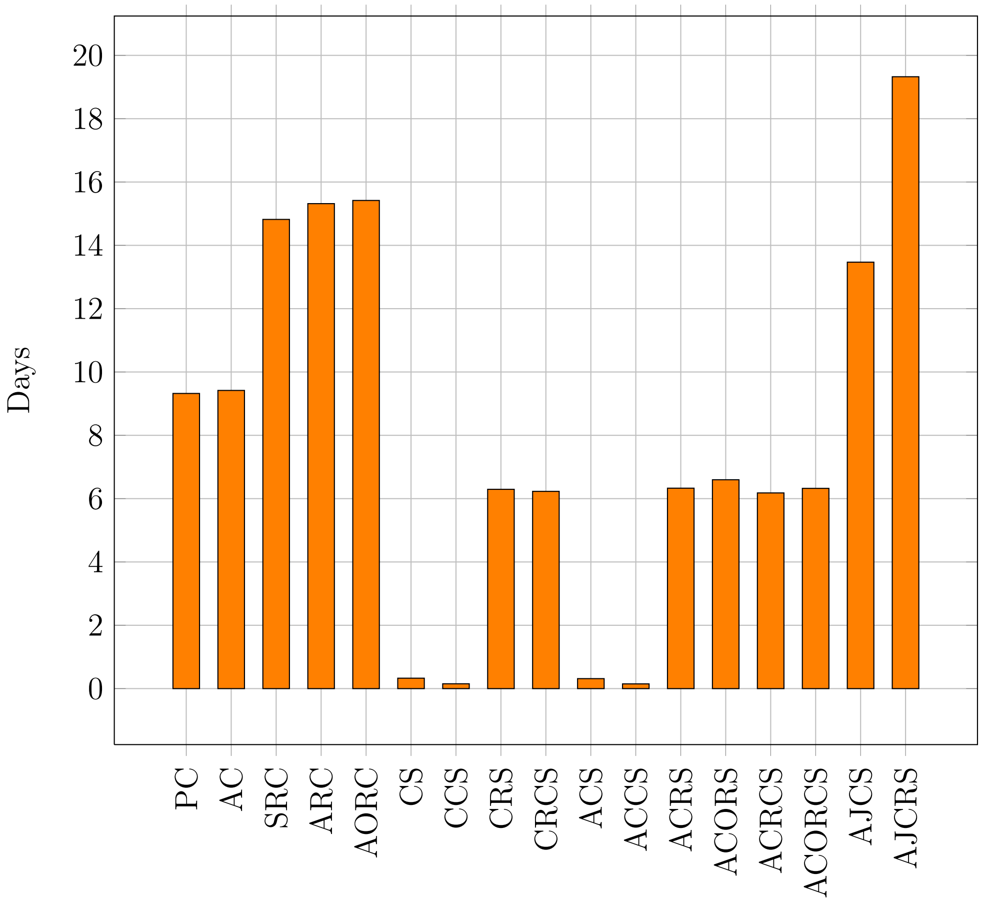


Figure 5.23: Calculation durations of each algorithm by taking the user-based approach into account and by the usage of the dataset from MovieLens

In contrast to the results that are presented above, Figure 5.24 and Figure 5.25 present the results that use an active user/item for the calculation. The accomplished experiment predicts only the entries of this active user/item and not the entire entries of the user-item matrix.

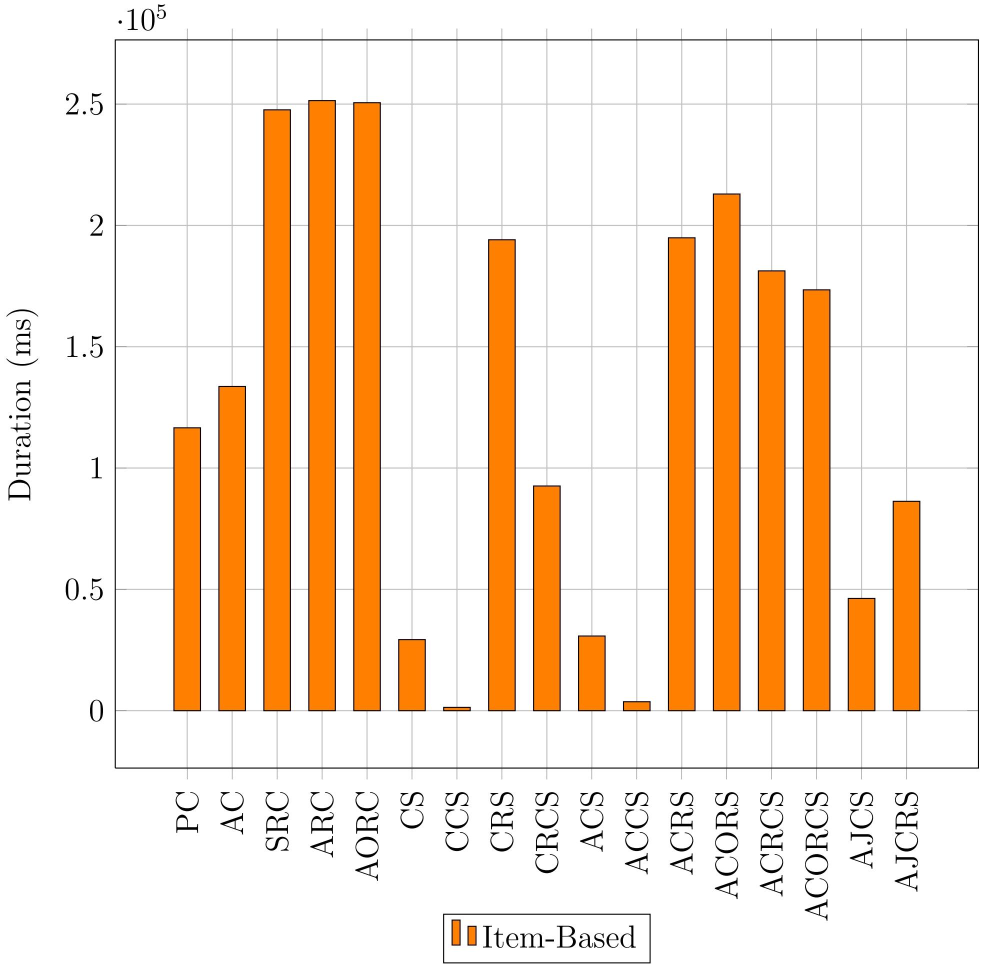


Figure 5.24: Calculation durations of each algorithm by taking the item-based approach into account by the usage of the dataset from MovieLens and an active user

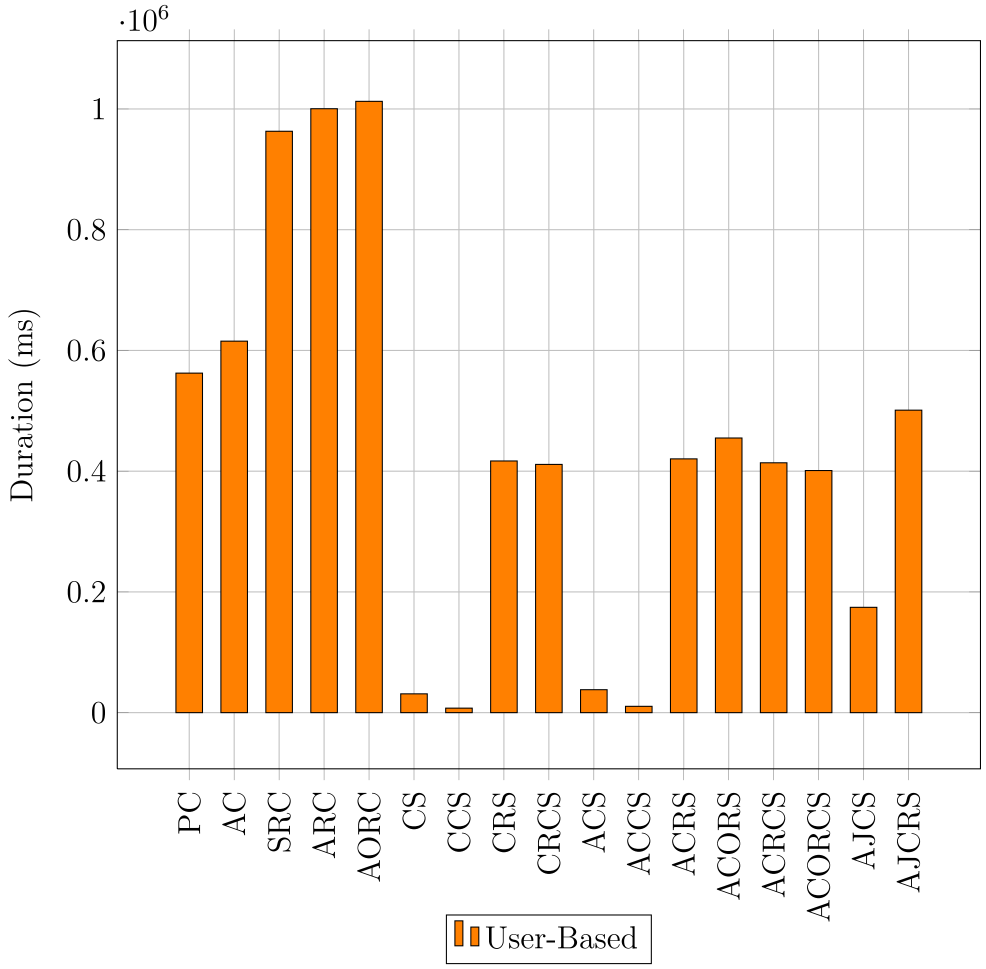


Figure 5.25: Calculation durations of each algorithm by taking the user-based approach into account by the usage of the dataset from MovieLens and an active user

5.2.3 Performance

This section presents the performance of the considered collaborative-filtering algorithms without the usage of the k-nearest neighbour approach. The tests were performed with a simulation test. This test built randomly filled user-item matrices with values. The tests include twelve items, which represent the twelve main genres that are specified by ETSI. The tests were accomplished with 10,20,...,200 users and 12 items. The results present the duration of the calculation of each algorithm within the proposed system. The tests were performed with a CPU of 2Ghz.

Figure 5.26 presents the results in a single view. The results show that the duration of the calculations increases exponentially.

Figures 5.27, 5.28, 5.29, and 5.30 present a selection of all algorithms which are split by the duration of the calculations.

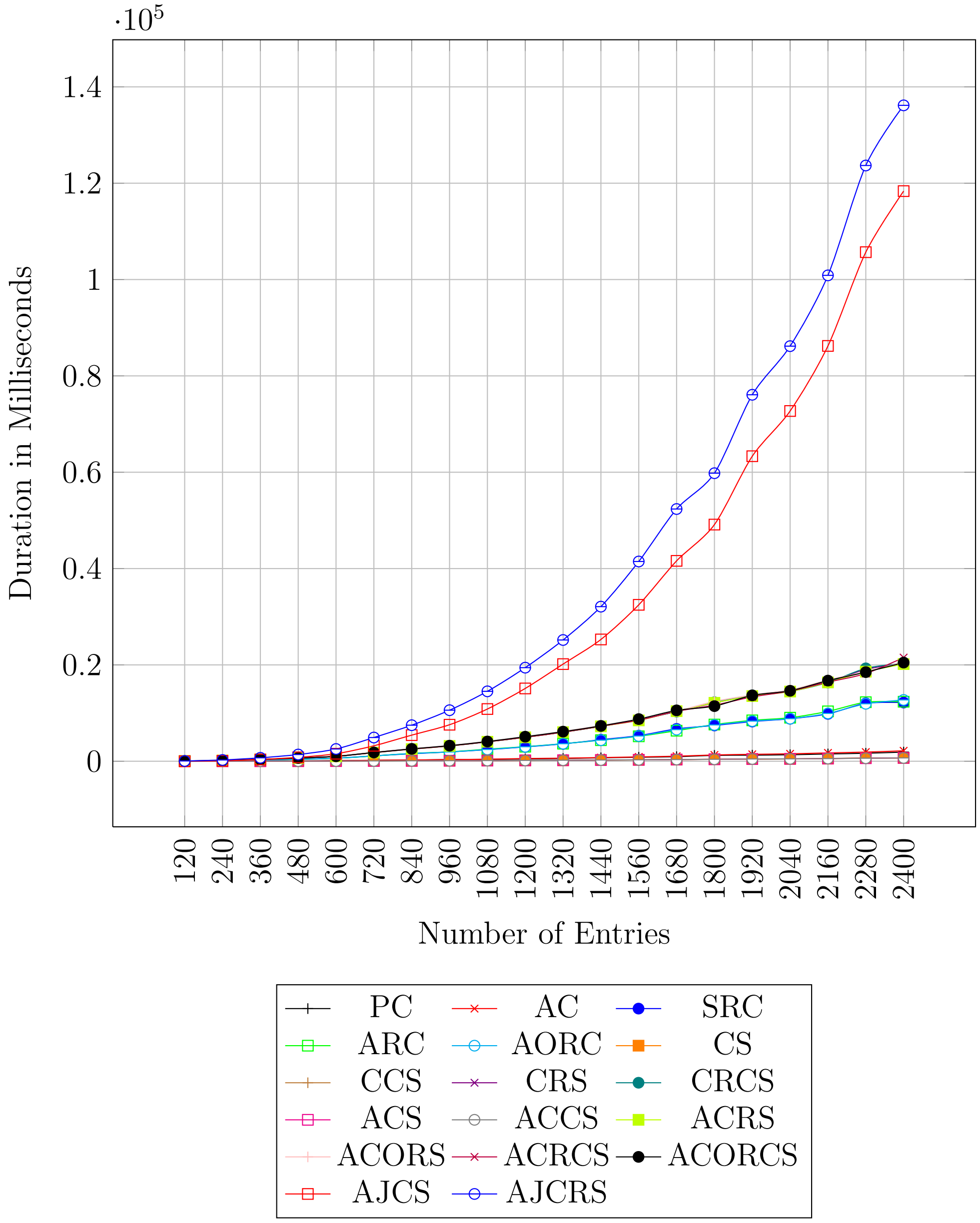


Figure 5.26: Calculation durations of the single collaborative-filtering algorithms without considering the neighbourhood

Figure 5.27 shows that the algorithms, which are based on the *Cosine Similarity* are the fastest algorithms.

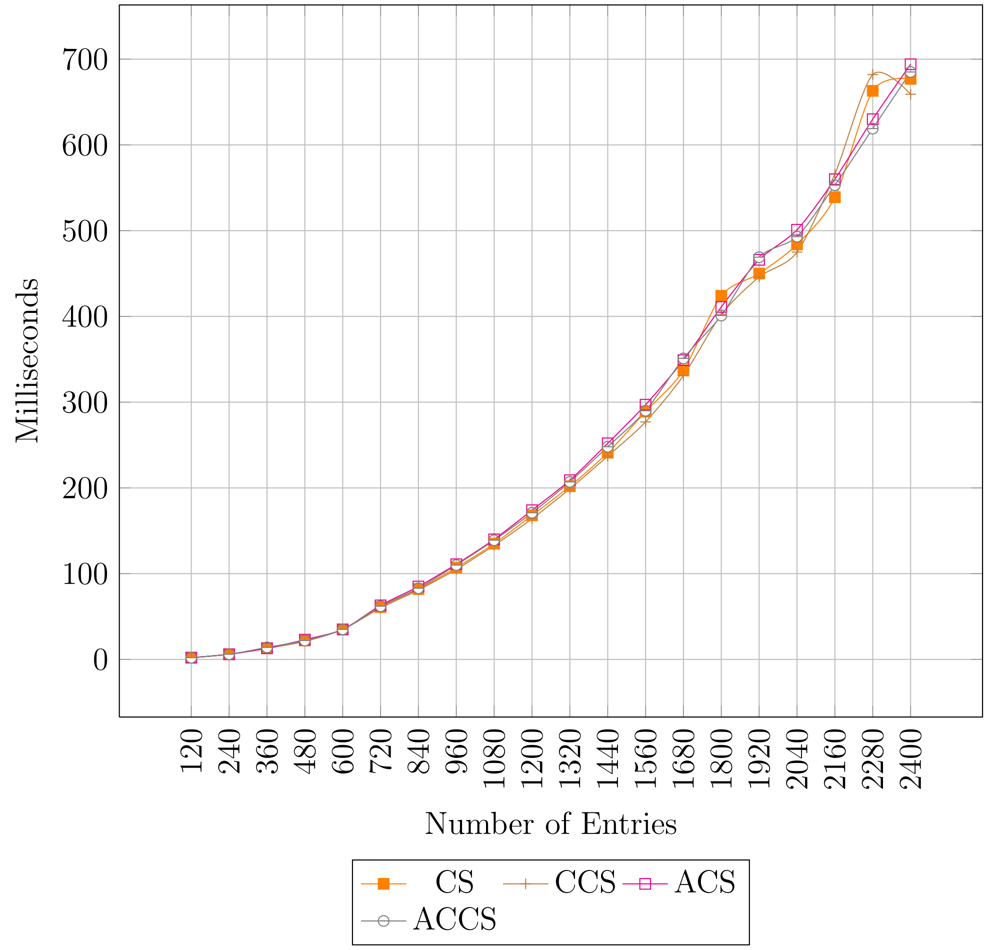


Figure 5.27: Calculation durations of the single collaborative-filtering algorithms without considering the neighbourhood which are very fast compared to the other algorithms

Figure 5.28 shows that the *Pearson-r Correlation* and the *Absolute Correlation* are the second fastest.

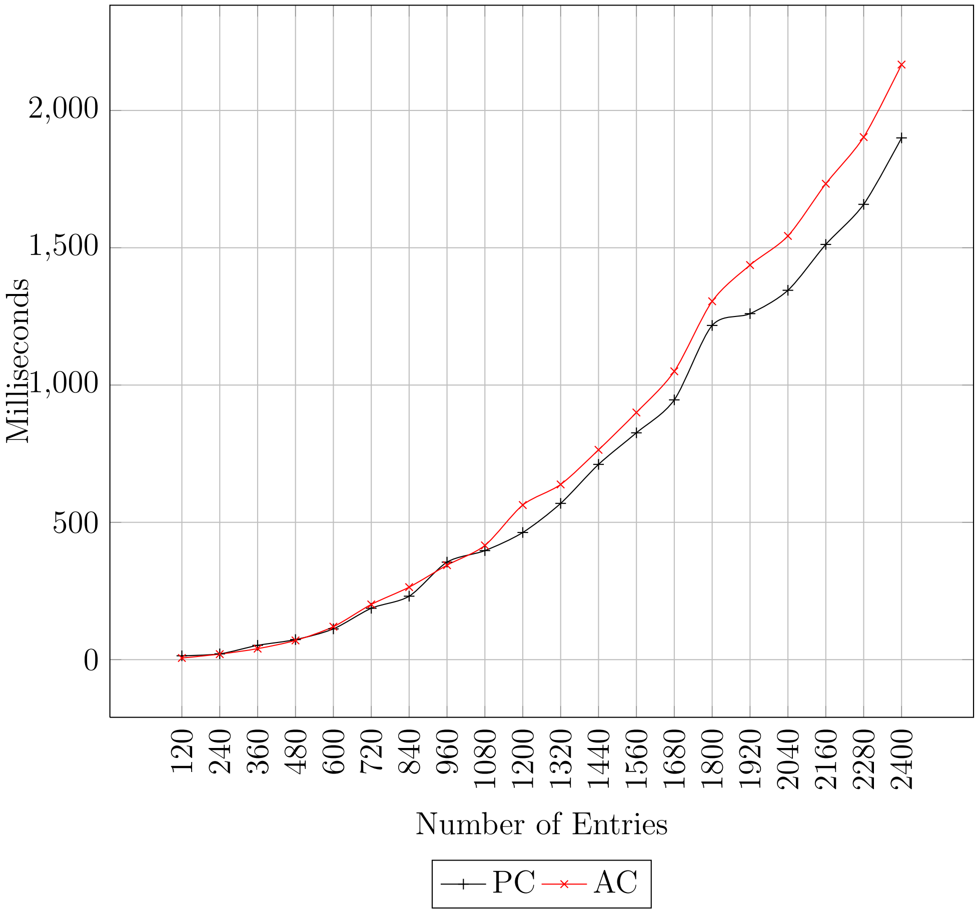


Figure 5.28: Calculation durations of the single collaborative-filtering algorithms without considering the neighbourhood which are fast compared to the other algorithms

Figure 5.29 shows that the algorithms which are based on the building of ranks are the third fastest. It also shows that the algorithms which use the basis of *Cosine Similarity* and the building of ranks are slower than the algorithms that are based on the *Pearson-r Correlation* and the *Spearman Rank Correlation*.

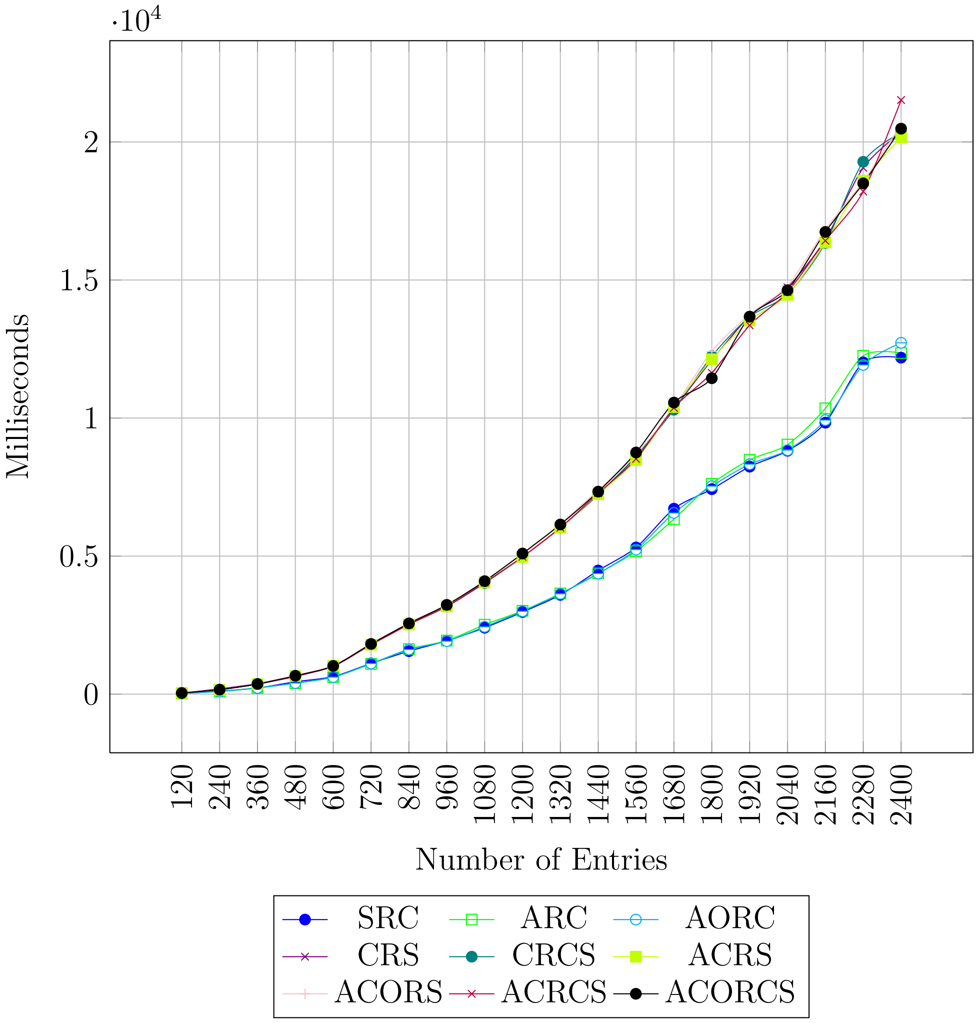


Figure 5.29: Calculation durations of the single collaborative-filtering algorithms without considering the neighbourhood which are slow compared to the other algorithms

Figure 5.30 shows that the *Adjusted Cosine Similarity* and the *Adjusted Cosine Rank Similarity* are the slowest algorithms.

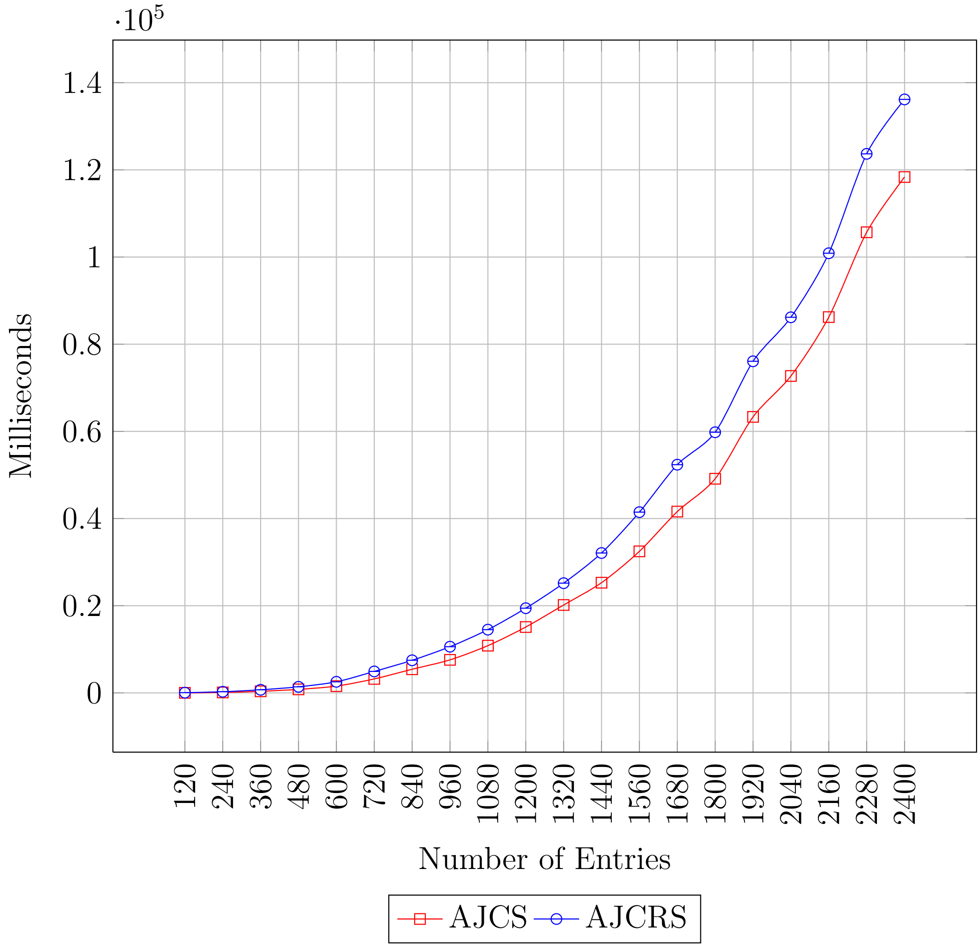


Figure 5.30: Calculation durations of the single collaborative-filtering algorithms without considering the neighbourhood which are very slow compared to the other algorithms

Table 5.2 describes the functions of each algorithm. The functions of each algorithm have been exploited by using the linear regression approach. Each function is described with a polynom of the third grade.

Algorithm	**f(x)**
PC	$f(x) = -0.085 \cdot x^3 + 6.999 \cdot x^2 - 14.284 \cdot x + 25.285$
AC	$f(x) = -0.058 \cdot x^3 + 7.083 \cdot x^2 - 12.405 \cdot x + 14.974$
SRC	$f(x) = -0.533 \cdot x^3 + 47.959 \cdot x^2 - 132.295 \cdot x + 185.953$
ARC	$f(x) = -0.387 \cdot x^3 + 45.363 \cdot x^2 - 120.55 \cdot x + 170.206$
AORC	$f(x) = -0.176 \cdot x^3 + 38.45 \cdot x^2 - 63.812 \cdot x + 65.869$
CS	$f(x) = -0.017 \cdot x^3 + 2.282 \cdot x^2 - 4.799 \cdot x + 6.243$
CCS	$f(x) = -0.008 \cdot x^3 + 2.103 \cdot x^2 - 4.353 \cdot x + 6.542$
CRS	$f(x) = -0.617 \cdot x^3 + 71.523 \cdot x^2 - 166.845 \cdot x + 219.755$
CRCS	$f(x) = -0.567 \cdot x^3 + 69.931 \cdot x^2 - 152.907 \cdot x + 183.713$
ACS	$f(x) = -0.024 \cdot x^3 + 2.462 \cdot x^2 - 5.224 \cdot x + 6.196$
ACCS	$f(x) = -0.027 \cdot x^3 + 2.528 \cdot x^2 - 5.92 \cdot x + 7.534$
ACRS	$f(x) = -0.726 \cdot x^3 + 73.06 \cdot x^2 - 169.276 \cdot x + 202.399$
ACORS	$f(x) = -0.757 \cdot x^3 + 74.676 \cdot x^2 - 173.017 \cdot x + 205.528$
ACRCS	$f(x) = 0.283 \cdot x^3 + 45.204 \cdot x^2 + 33.094 \cdot x - 127.226$
ACORCS	$f(x) = -0.492 \cdot x^3 + 66.069 \cdot x^2 - 107.225 \cdot x + 95.088$
AJCS	$f(x) = 14.818 \cdot x^3 + 5.912 \cdot x^2 - 55.962 \cdot x + 63.078$
AJCRS	$f(x) = 13.077 \cdot x^3 + 104.499 \cdot x^2 - 443.534 \cdot x + 615.482$

Table 5.2: Performance of the single collaborative-filtering algorithms by presenting the function $f(x)$

5.2.4 Conclusion

5.2.4.1 Error Rates

Section 5.2 presents the results of the experiments that do not use the k-nearest neighbour approach or a dynamic selection of the most accurate collaborative-filtering algorithm. Figure 5.8 and Figure 5.14 present the error rates by taking several user-item matrices into account. These results that use the dataset from the survey prove that the most accurate algorithm is strongly connected to the considered user-item matrix.

Besides this fact, the results prove the usefulness of the newly developed collaborative-filtering algorithms. Figure 5.16-5.21 present the results that consider the dataset from MovieLens. These results affirm the usefulness of the newly developed collaborative-filtering algorithms.

For example, the algorithm which produces the lowest error rate by taking the item-based approach into account is the *Absolute Original Rank Correlation*. The algorithm with the lowest error rate that uses the user-based approach is the *Absolute Correlation*.

5.2.4.2 Performance

Section 5.2.2.1 presents the calculation durations by considering the dataset from MovieLens. The results are presented in Figure 5.22 and Figure 5.23. Each entry within the user-item matrix is predicted and the active user/item is not considered.

Figure 5.24 and Figure 5.25 present the calculation durations of the experiments that consider an active user/item. In contrast to the above mentioned experiments, these experiments only predict the entries of the active user/item and not the entire user-item matrix.

Beside the performance results that take the dataset from MovieLens into account, Section 5.2.3 presents the calculation duration of the single collaborative-filtering algorithms by using different simulated user-item matrices. The results have been achieved by predicting every entry within the used user-item matrix and presented in Figure 5.26 at one glance. The experiments consider different number of entries.

5.3 With Neighbourhood

This section presents the results that consider the k-nearest neighbour approach. It presents the error rates that have been archived by calculating the predictions of entries and the comparison between the prediction and the original value.

5.3.1 Survey

This section presents the results which have been achieved by using the dataset from the survey. The used user-item matrix includes ratings from 10 users and 12 items, which represent genres from the DVB Standard for Service Information.

5.3.1.1 Item-Based

Figure 5.31 presents the results which have been achieved by using the k-nearest neighbour approach. It presents the results calculated with a neighbourhood size of three. The results are calculated by averaging the errors of ten test cycles.

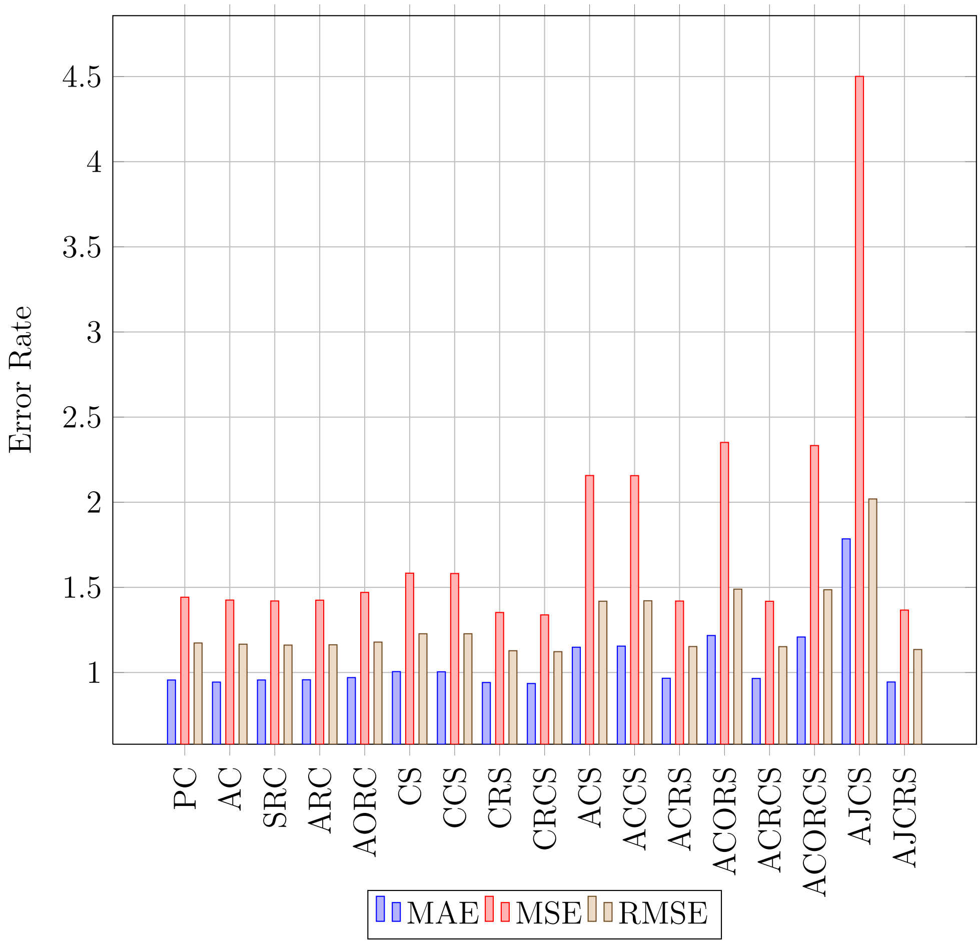


Figure 5.31: Error rates by considering the dataset from the survey by using the item-based approach and taking 3-nearest neighbours into account

5.3.1.2 User-Based

Figure 5.32 presents the error rates of the single algorithms by using the 3-nearest neighbours, while the active user also belongs to the neighbourhood. The results are the average of ten test cycles.

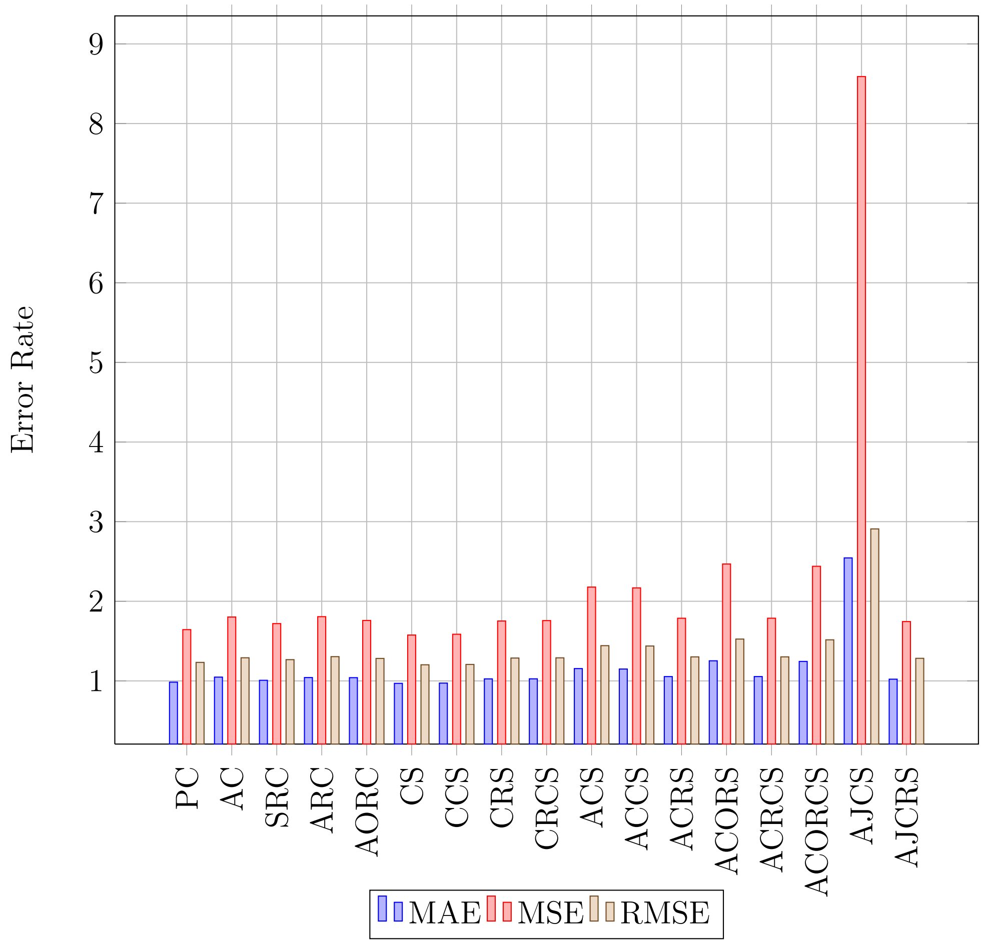


Figure 5.32: Error rates by considering the dataset from the survey by using the user-based approach and taking 3-nearest neighbours into account

5.3.2 MovieLens

This section considers the dataset from MovieLens which contains ratings from 943 users and 1682 items (movies). This dataset represents a huge community.

5.3.2.1 Item-Based

The error rates that take the item-based approach into account are presented in Figures 5.33-5.35. They present the averaged error rates by considering 10 test cycles. The figures also show the error rates that take different neighbourhood sizes into account. The tests are performed by using 10, 20, 30, 40, and 50 k-nearest neighbours. Figure 5.33 presents the calculated MAE, Figure 5.34 presents the calculated MSE, and Figure 5.35 presents the calculated RMSE.

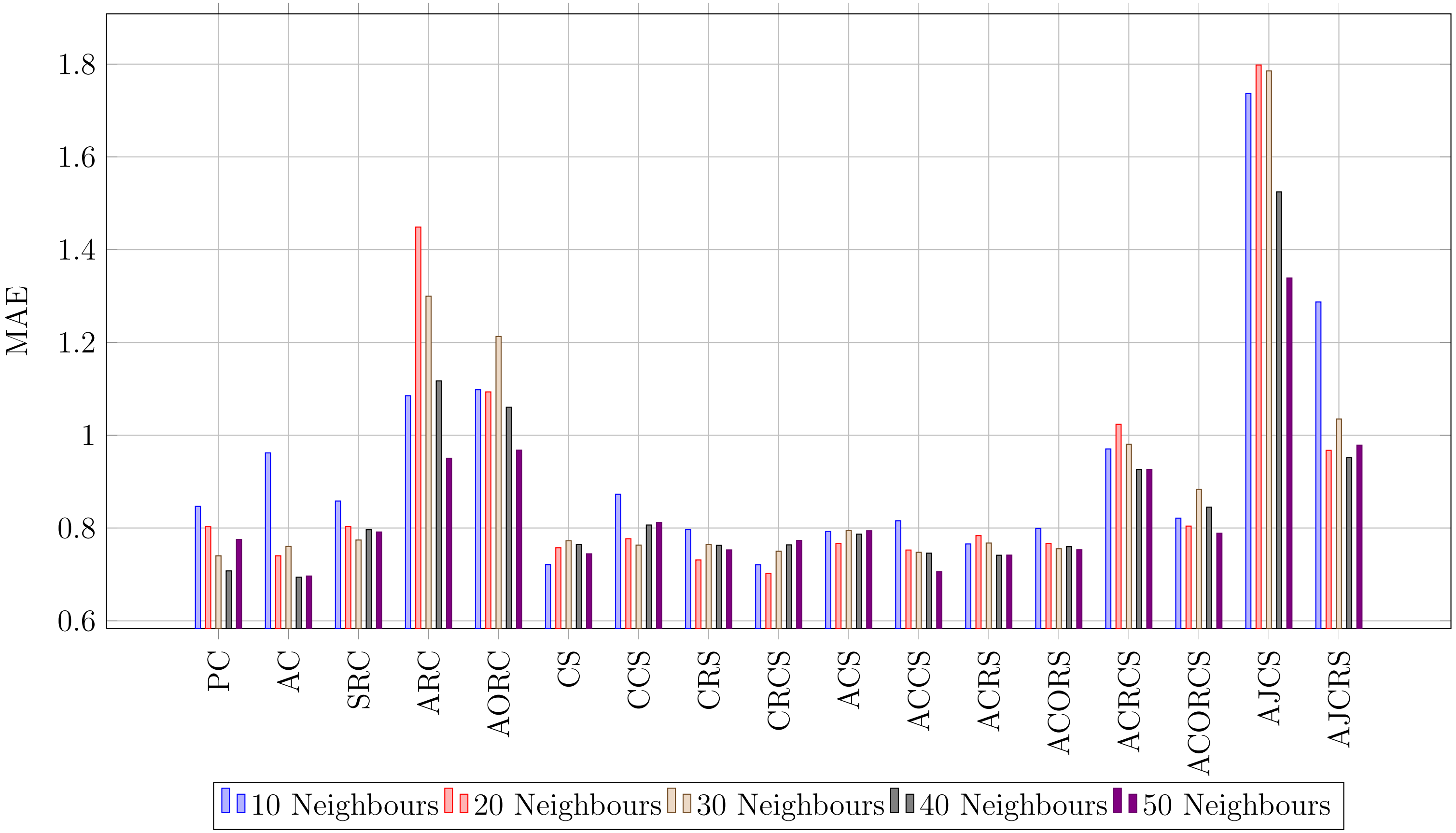


Figure 5.33: MAE by considering the item-based approach, taking 10-50 neighbours into account, performing 10 test cycles and using the dataset from MovieLens

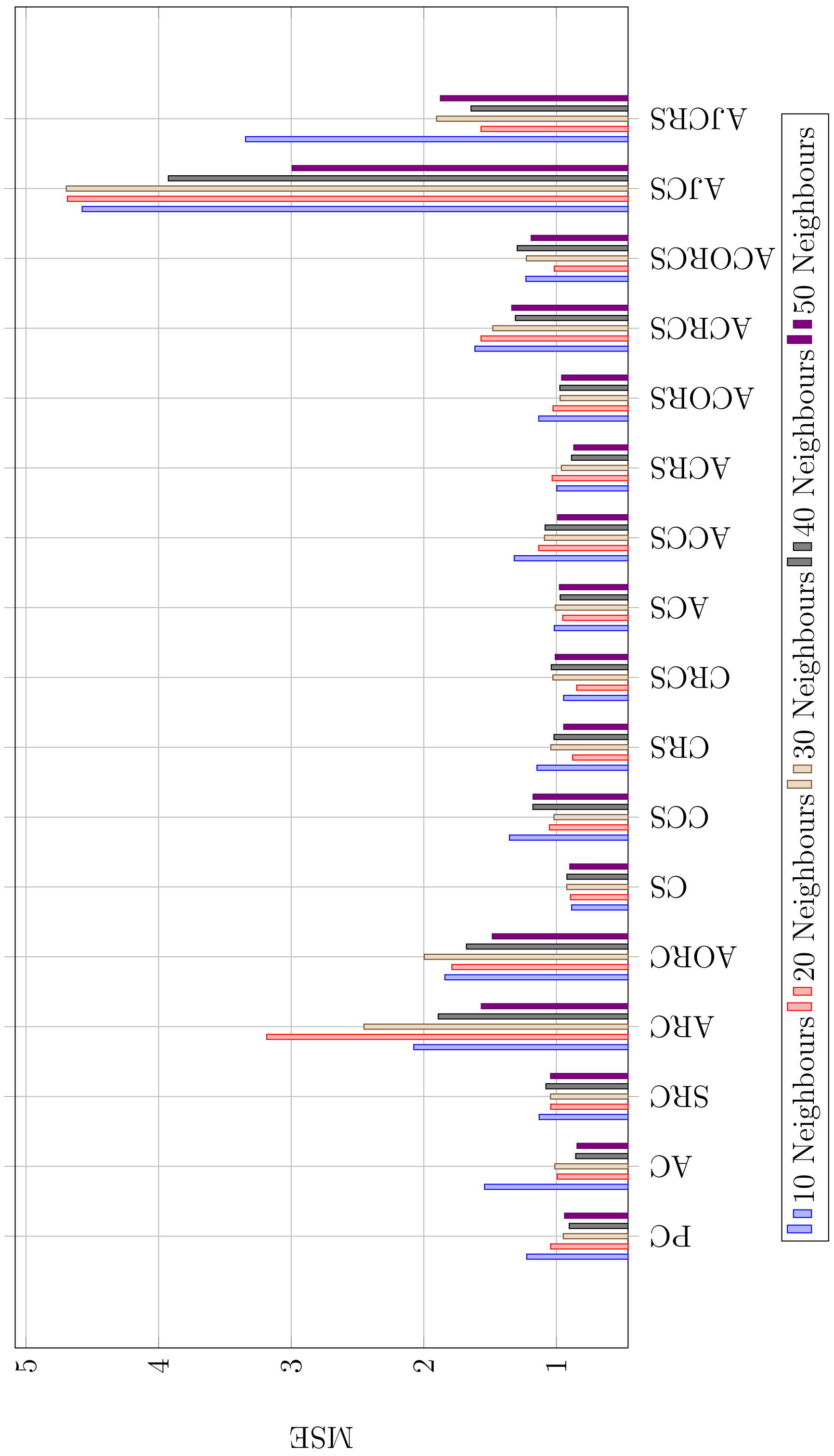


Figure 5.34: MSE by considering the item-based approach, taking 10-50 neighbours into account, performing 10 test cycles and using the dataset from MovieLens

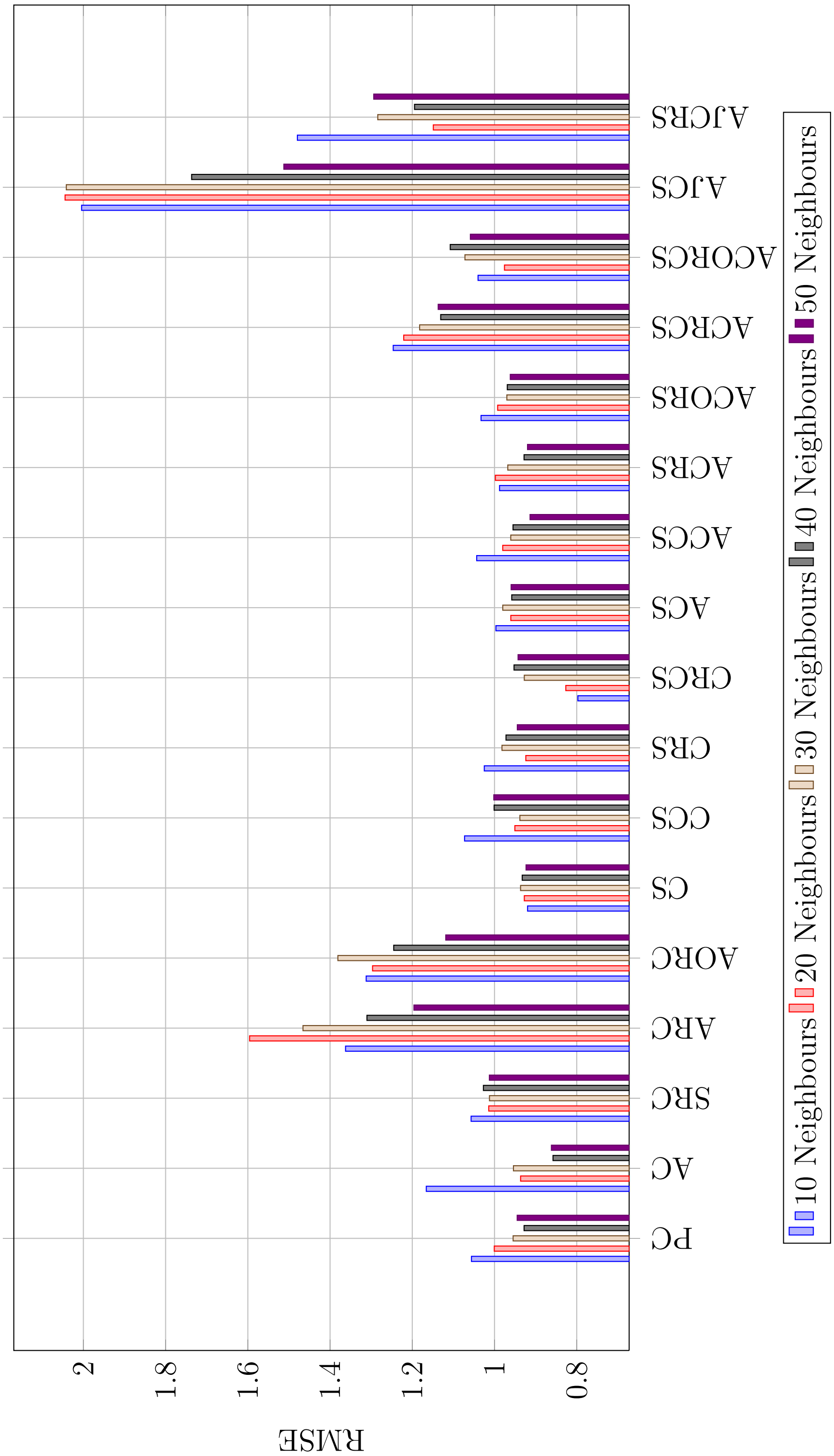


Figure 5.35: RMSE by considering the item-based approach, taking 10-50 neighbours into account, performing 10 test cycles and using the dataset from MovieLens

The error rates that take the item-based approach into account are presented in Figures 5.36-5.38. They present the averaged error rates by considering 50 test cycles. The figures also show the error rates that take different neighbourhood sizes into account. The tests were performed by using 10, 20, 30, 40, and 50 k-nearest neighbours. Figure 5.36 presents the calculated MAE, Figure 5.37 presents the calculated MSE, and Figure 5.38 presents the calculated RMSE.

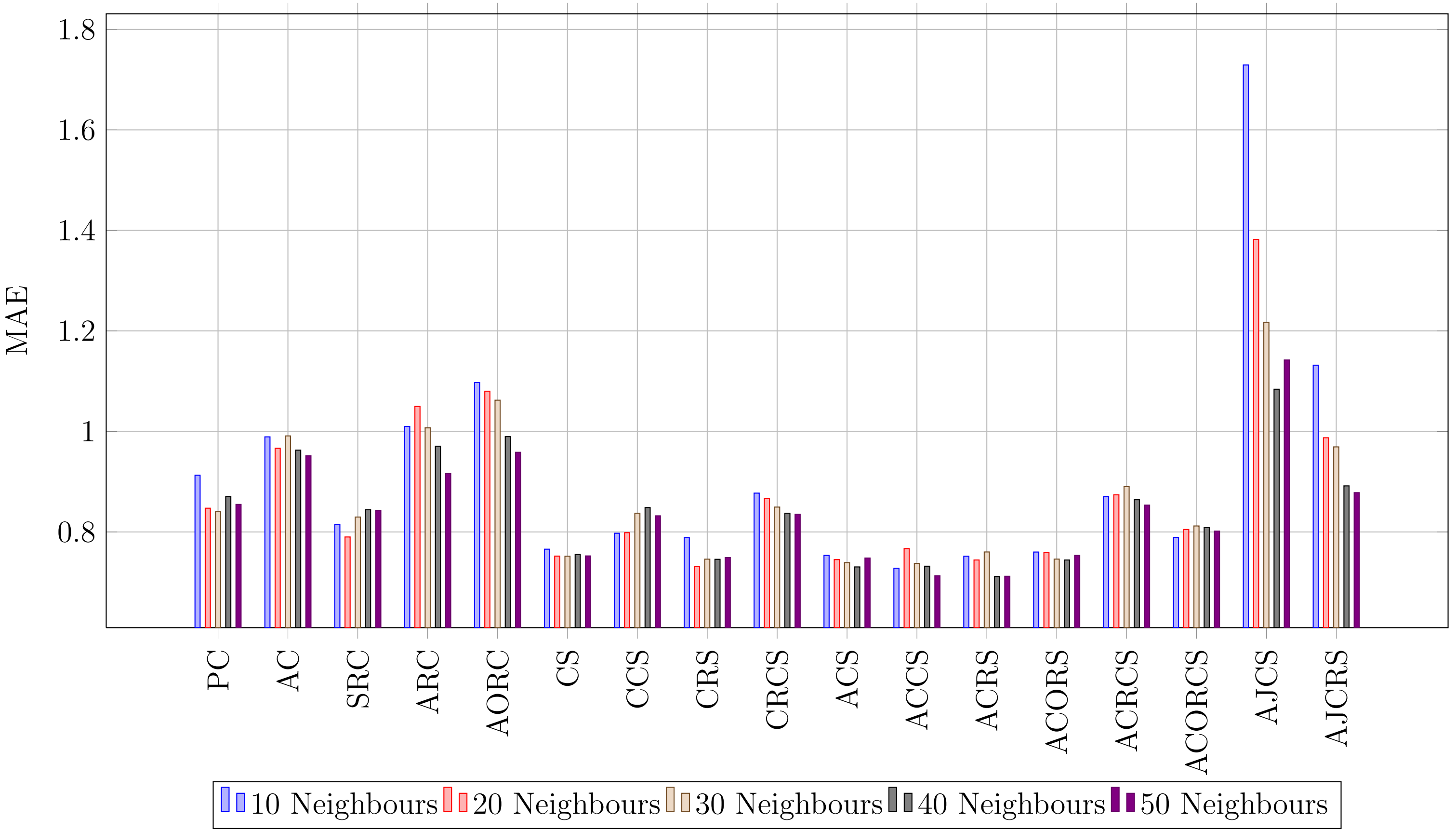


Figure 5.36: MAE by considering the item-based approach, taking 10-50 neighbours into account, performing 50 test cycles and using the dataset from MovieLens

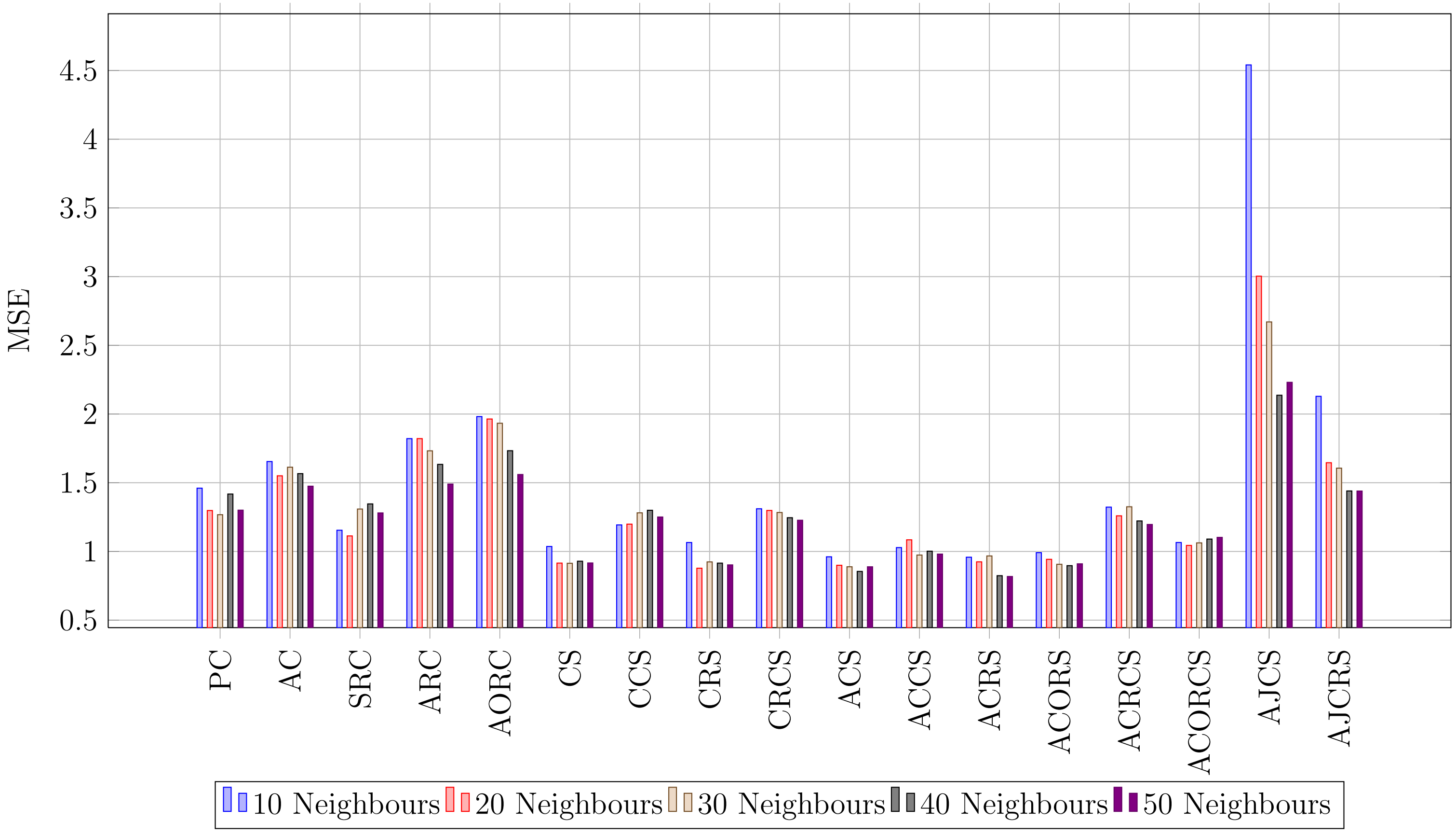


Figure 5.37: MSE by considering the item-based approach, taking 10-50 neighbours into account, performing 50 test cycles and using the dataset from MovieLens

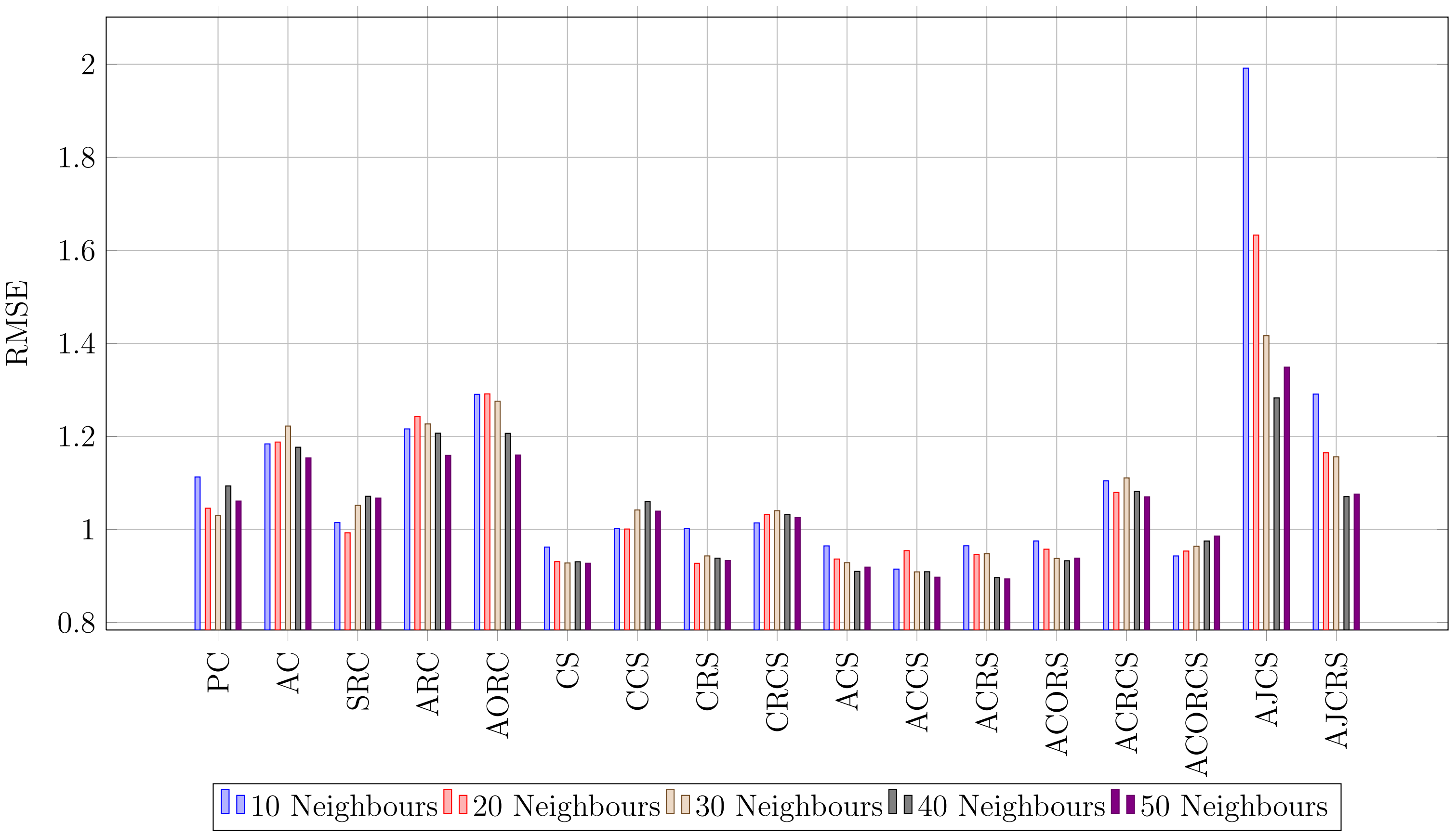


Figure 5.38: RMSE by considering the item-based approach, taking 10-50 neighbours into account, performing 50 test cycles and using the dataset from MovieLens

5.3.2.2 User-Based

Figures 5.39-5.41 present the error rates by using the user-based approach. The presented results consider different k-nearest neighbour sizes, such as 10, 20, 30, 40, and 50 k-nearest neighbours. The tables also show the averaged error rates from 10 test cycles.

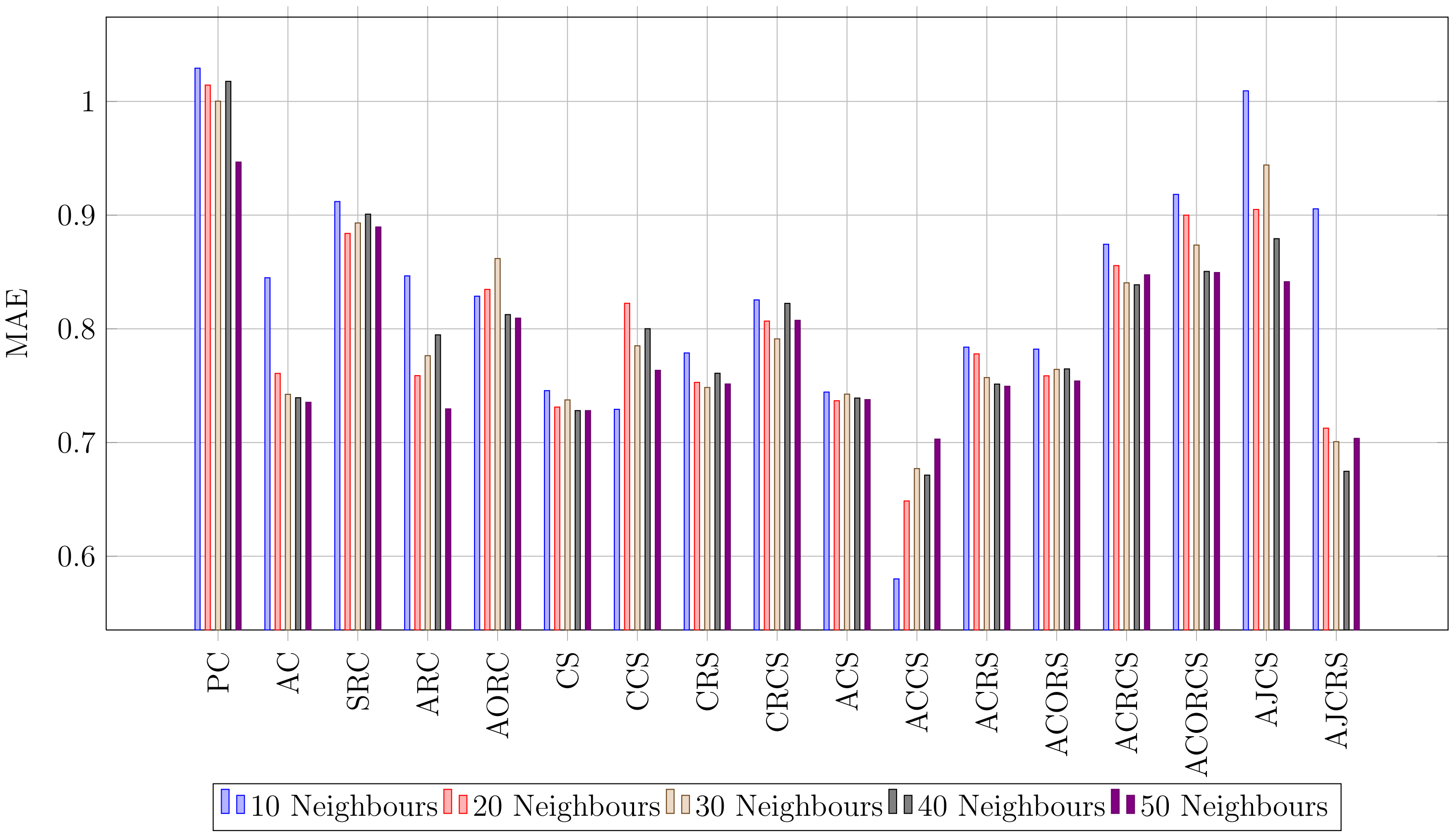


Figure 5.39: MAE by considering the user-based approach, taking 10-50 neighbours into account, performing 10 test cycles and using the dataset from MovieLens

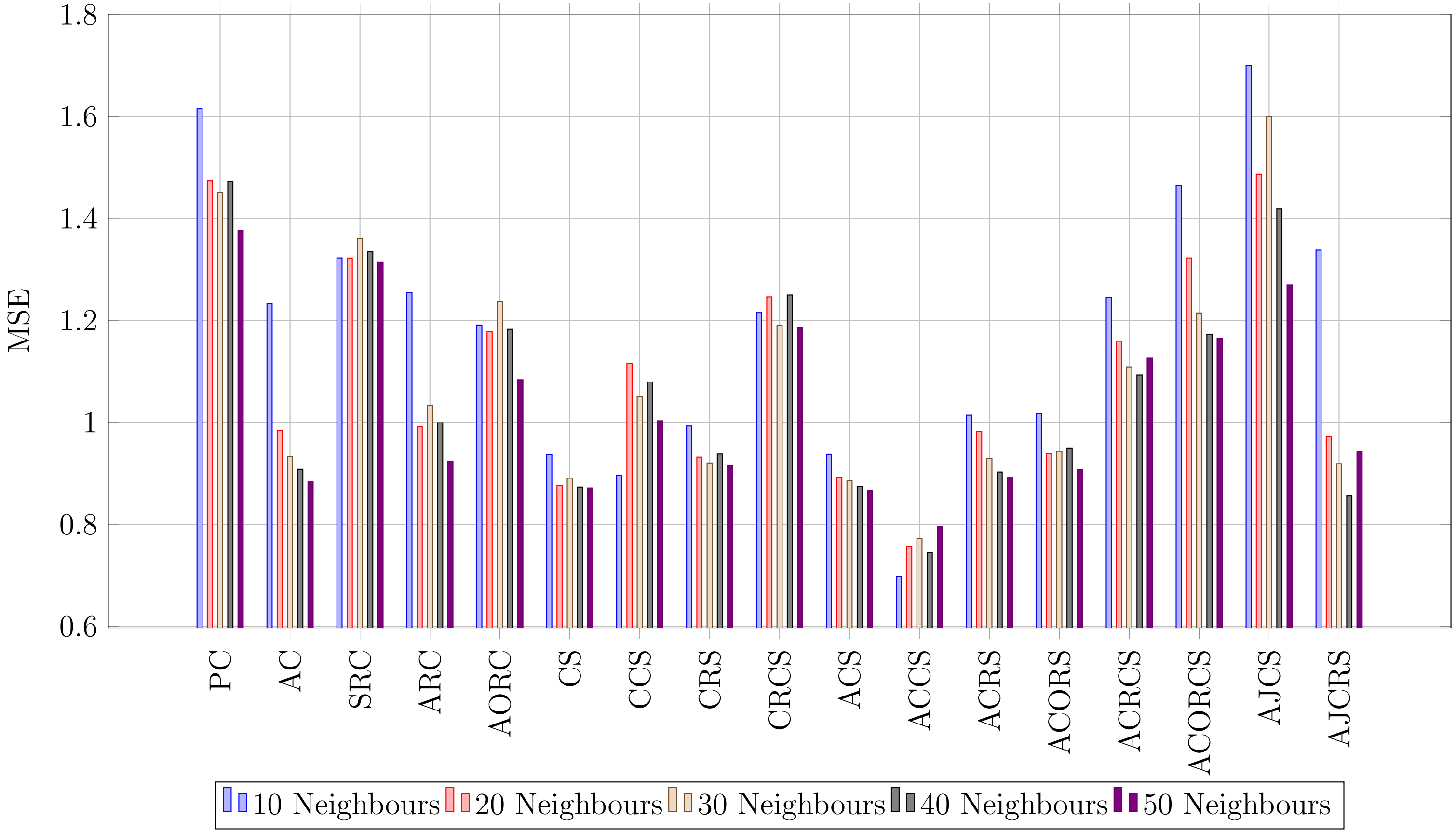


Figure 5.40: MSE by considering the user-based approach, taking 10-50 neighbours into account, performing 10 test cycles and using the dataset from MovieLens

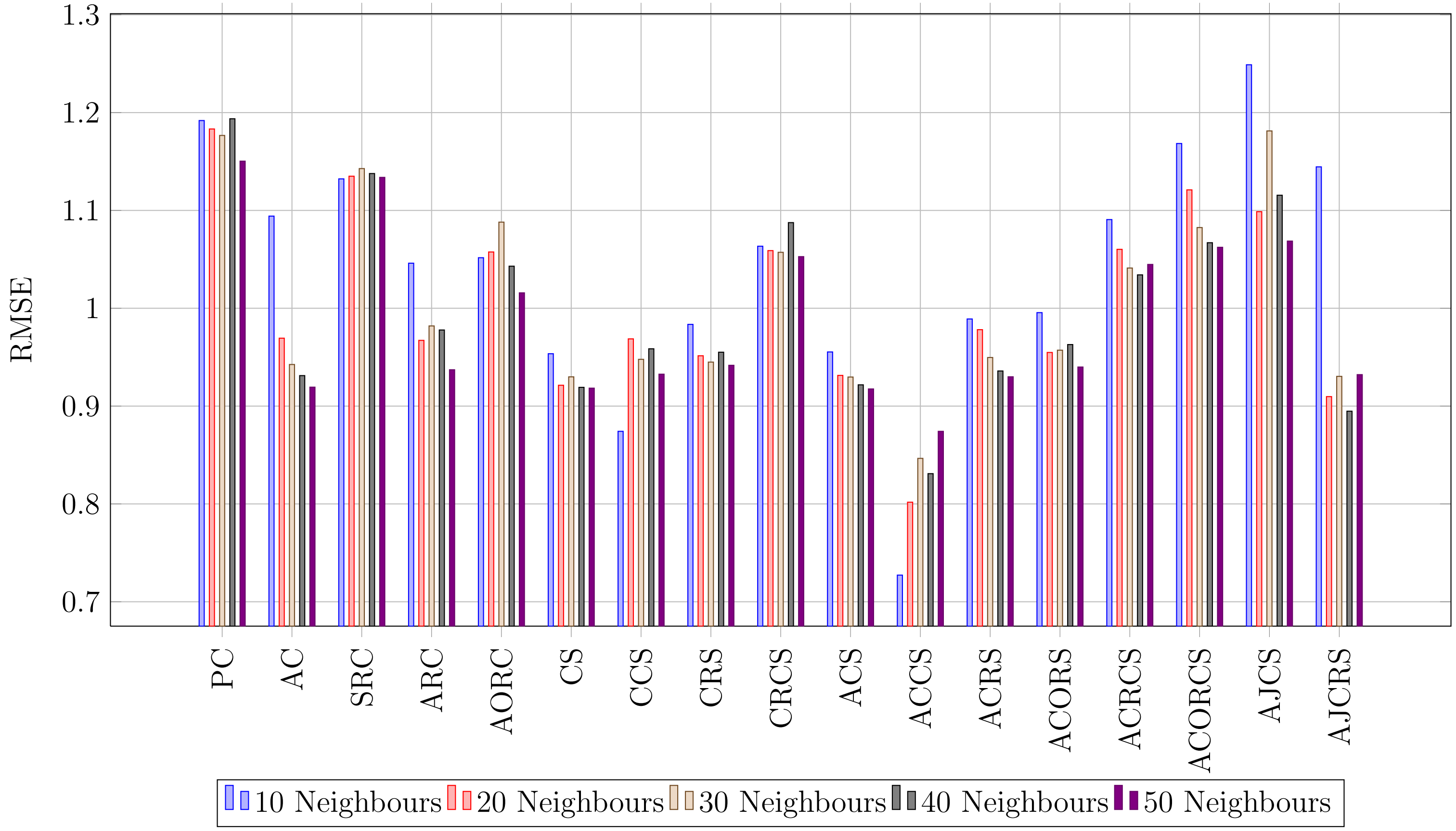


Figure 5.41: RMSE by considering the user-based approach, taking 10-50 neighbours into account, performing 10 test cycles and using the dataset from MovieLens

Figures 5.42-5.44 present the error rates by using the user-based approach. The presented results consider different k-nearest neighbour sizes, such as 10, 20, 30, 40, and 50 k-nearest neighbours. The tables also show the averaged error rates from 50 test cycles.

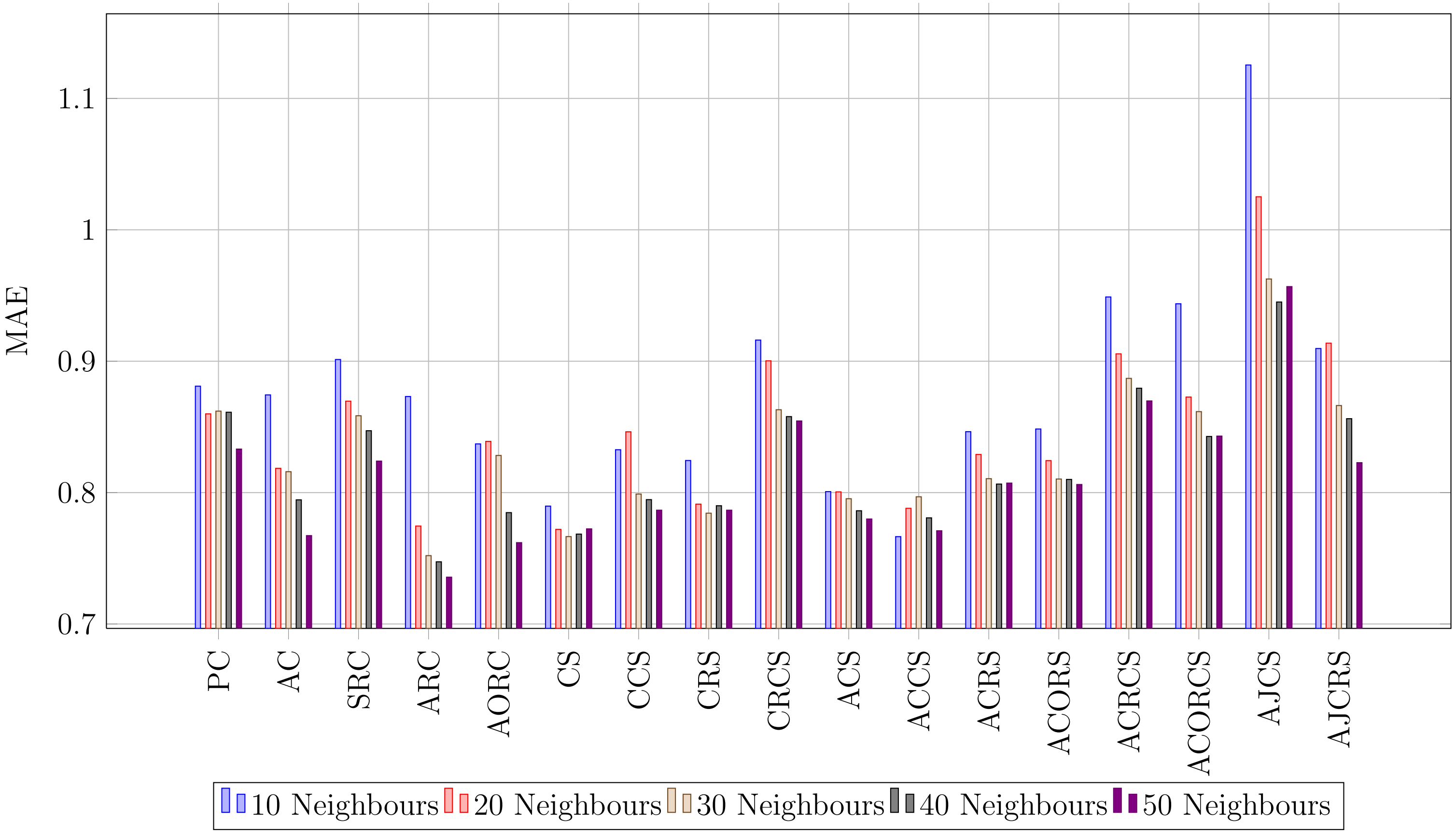


Figure 5.42: MAE by considering the user-based approach, taking 10-50 neighbours into account, performing 50 test cycles and using the dataset from MovieLens

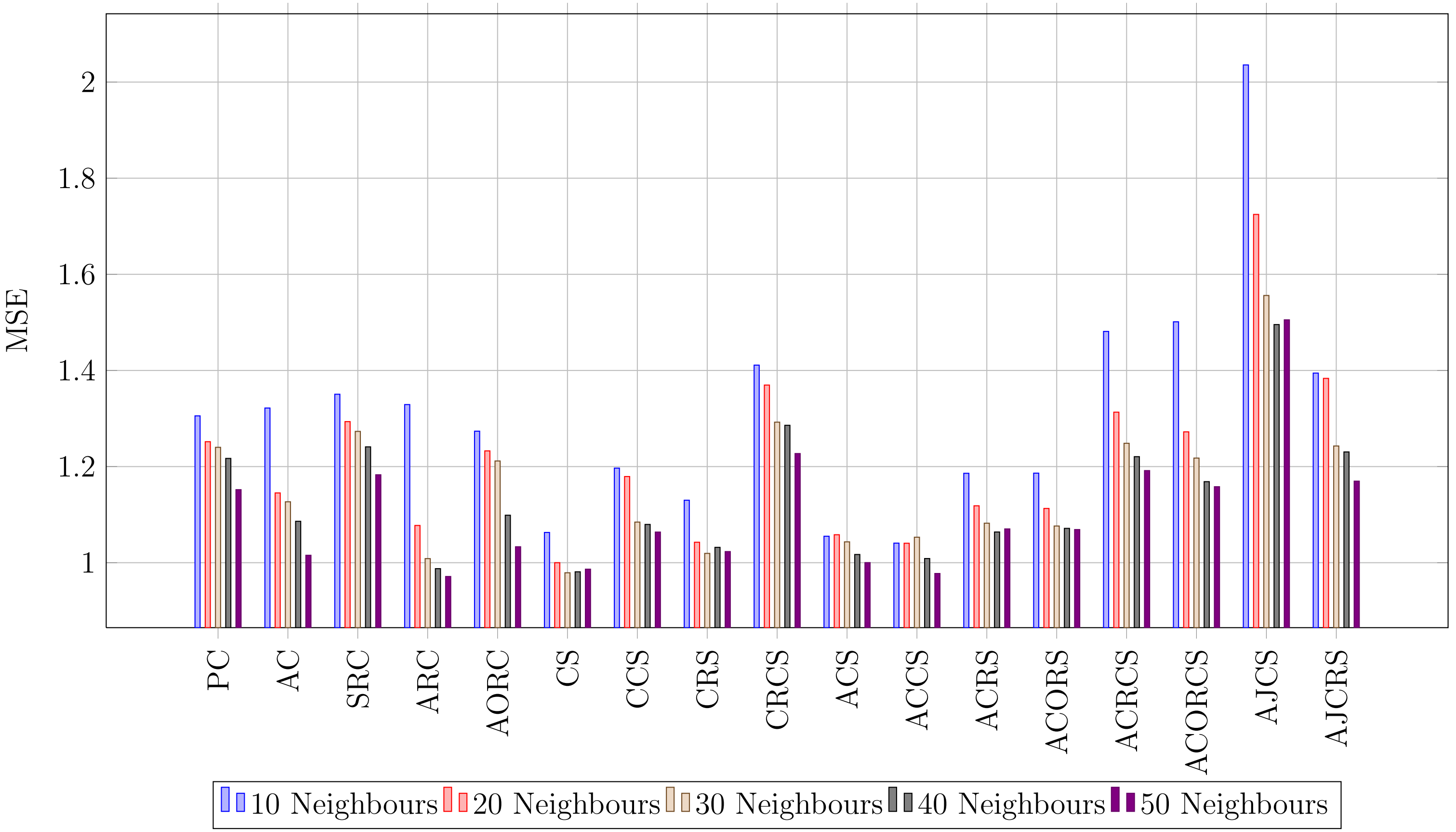


Figure 5.43: MSE by considering the user-based approach, taking 10-50 neighbours into account, performing 50 test cycles and using the dataset from MovieLens

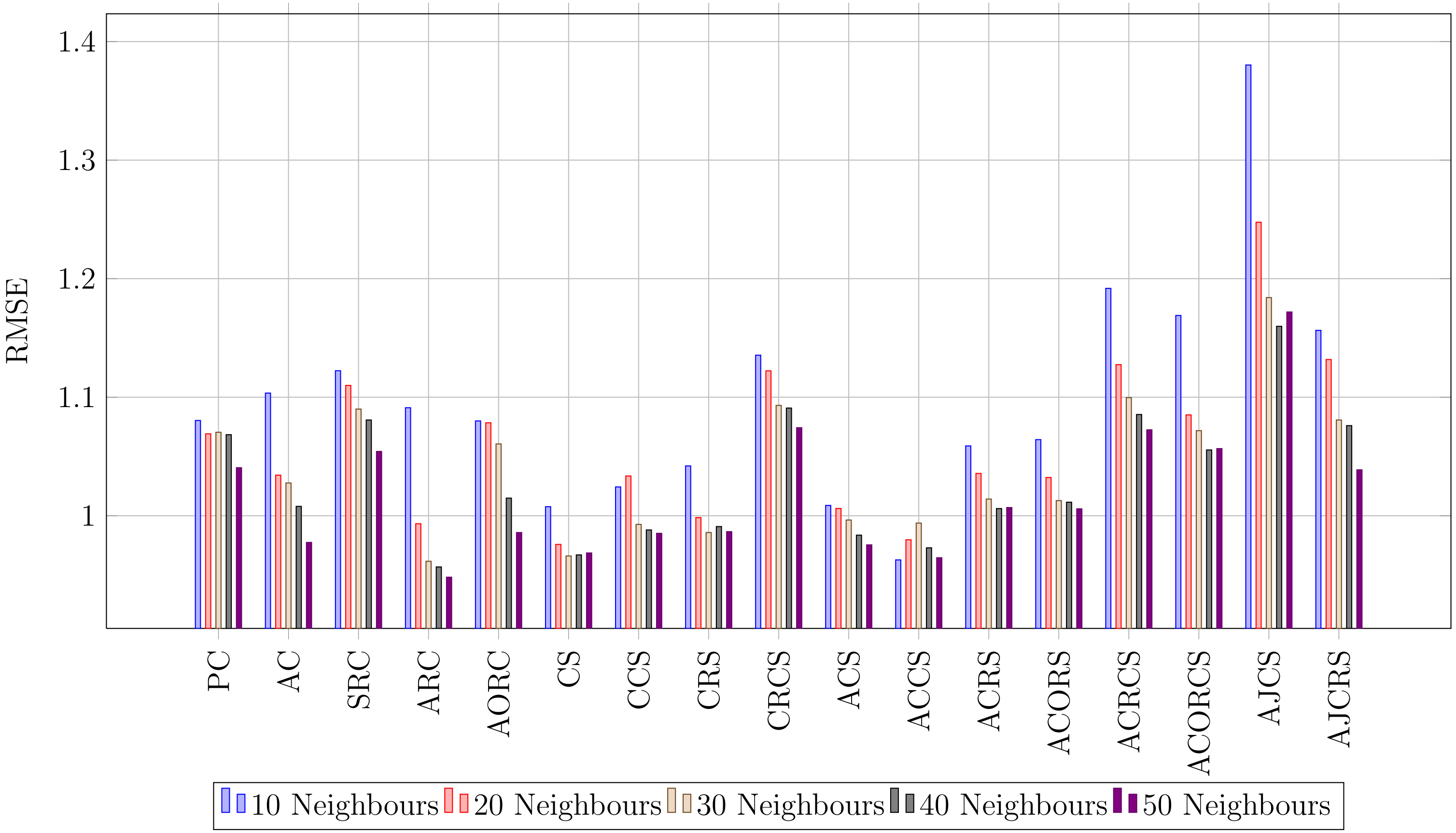


Figure 5.44: RMSE by considering the user-based approach, taking 10-50 neighbours into account, performing 50 test cycles and using the dataset from MovieLens

5.3.2.3 Performance

This section presents the calculation durations of the single collaborative-filtering algorithms by considering different neighbourhood sizes. The table presents the results by taking 10, 20, 30, 40, and 50 neighbours into account. The measurement unit is milliseconds (ms). Figure 5.45 presents the results which used the item-based approach. The results of the calculation durations that considers the user-based approach are presented in Figure 5.46.

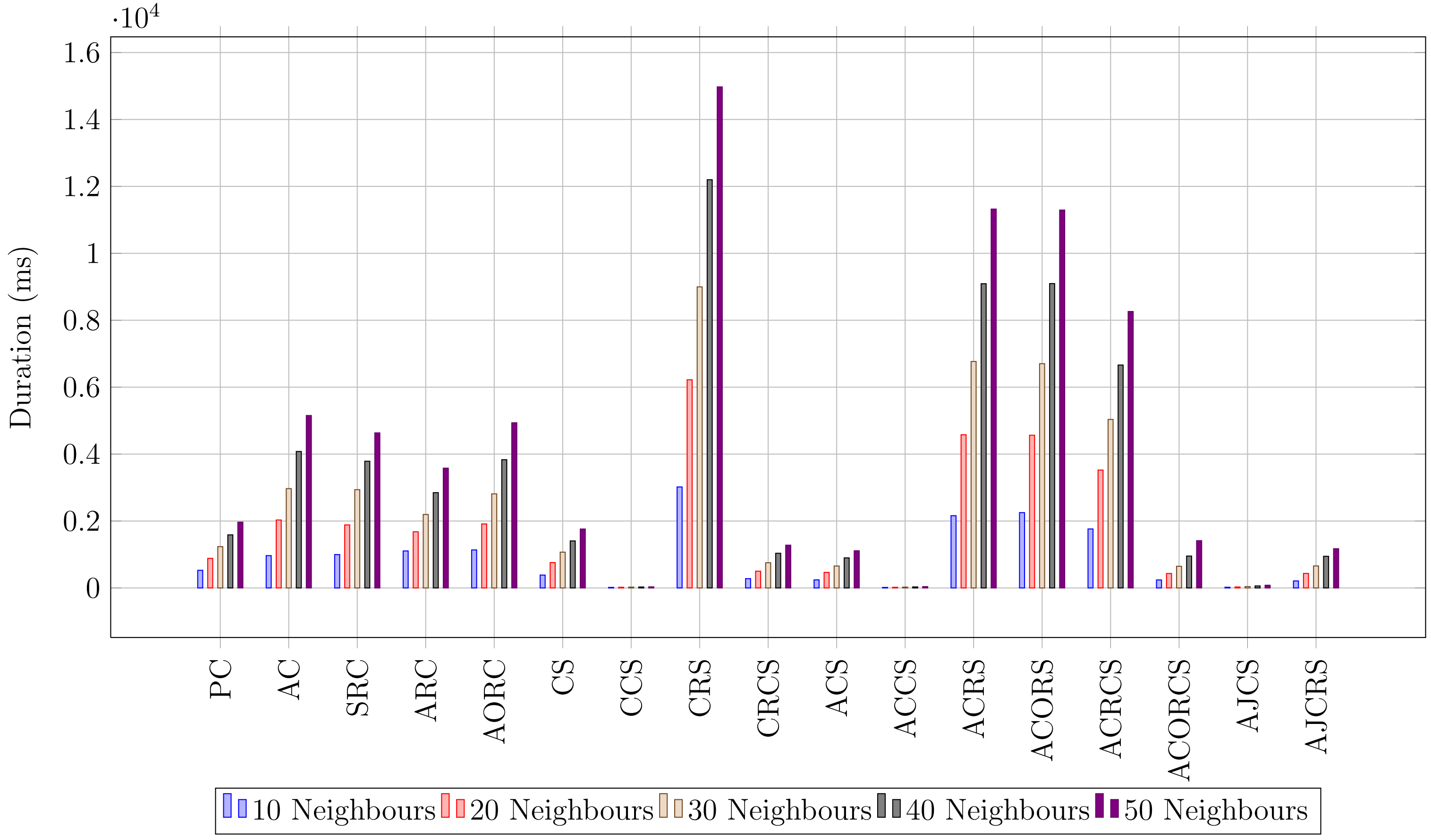


Figure 5.45: Calculation duration by considering 10, 20, 30, 40, and 50 k-nearest neighbours by taking the item-based approach into account

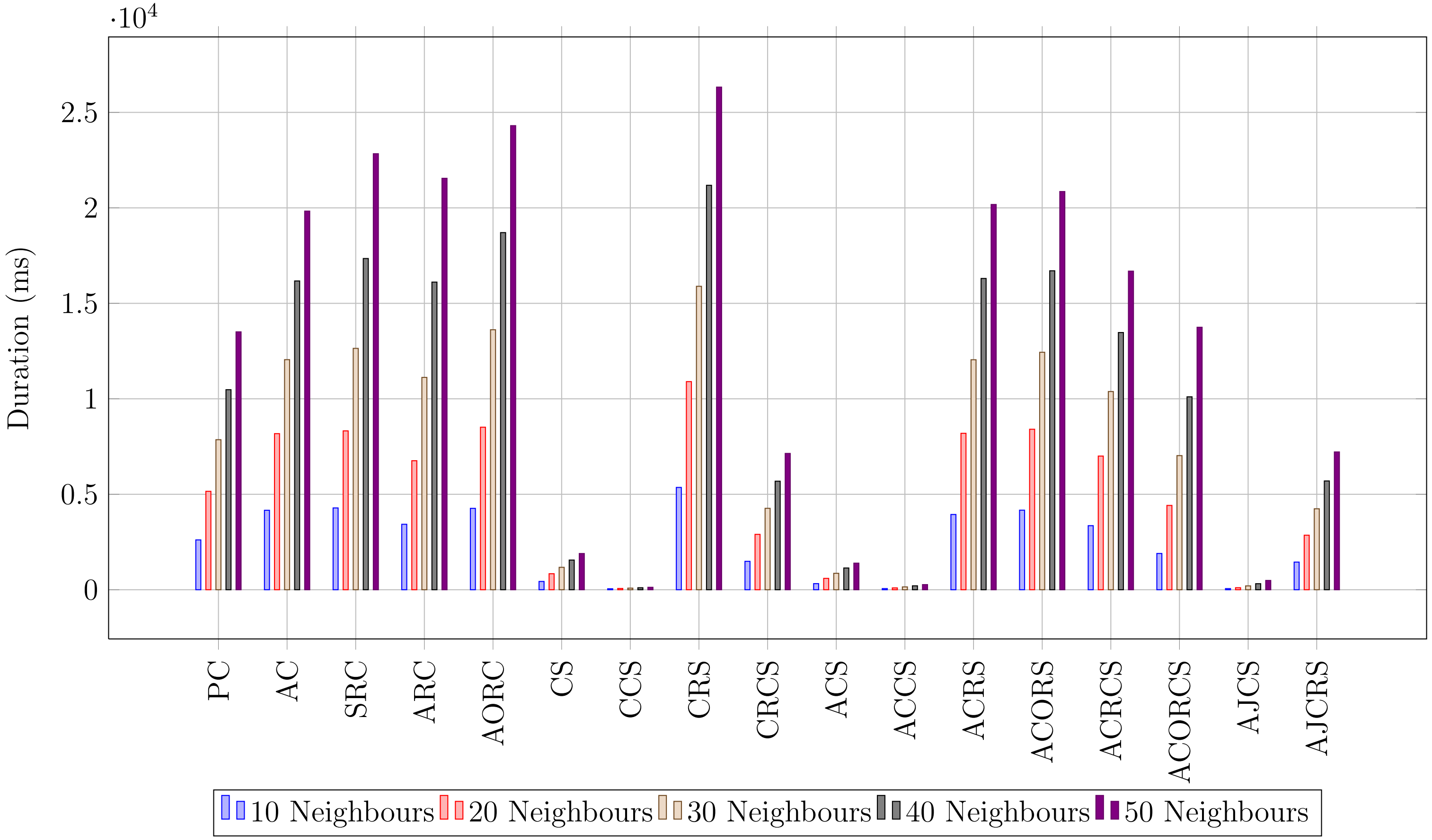


Figure 5.46: Calculation duration by considering 10, 20, 30, 40, and 50 k-nearest neighbours by taking the user-based approach into account

5.3.3 Conclusion

5.3.3.1 Error Rates

Section 5.3 presents the results of the experiments that use the k-nearest neighbour approach. The results prove that the creation of a neighbourhood is able to reduce the error rates compared to the approach, which does not use this neighbourhood.

For example, Figure 5.31 and Figure 5.32 present the error rates that have been achieved by using a neighbourhood size of three and taking the dataset from the survey into account. These error rates are lower than the results from the experiment that do not use the k-nearest neighbour approach. The experiments which use the dataset from MovieLens also prove the improvement of the predictions' accuracy by using the k-nearest neighbour approach.

Additionally the results also affirm that in most cases the newly developed collaborative-filtering algorithms are able to produce a lower error rate than SotA algorithms.

5.3.3.2 Performance

Section 5.3.2.3 presents the performance results that have been achieved by considering the dataset from MovieLens. Figure 5.45 presents the results which takes the item-based approach into account. Figure 5.46 presents the results that uses the user-based approach. The tables show the calculation duration by using the neighbourhood sizes of 10, 20, 30, 40, and 50 neighbours. In contrast to the experiments that do not use the neighbourhood approach and the active user, the calculation duration that considers a neighbourhood is significantly lower.

5.4 Dynamic Selection

This section presents the results that considers the proposed dynamic multi-algorithm collaborative-filtering system. The results prove the usefulness of this system and show that the error rates are significantly lower than under existing approaches.

5.4.1 Showing the Need

Figure 5.47 presents the MAE by using the dataset from MovieLens and taking different active users into account. The figure presents the results of five different active users. User 0, User 100, User 150, User 200, and User 250 were set to the active user. The neighbourhood size of these results is 10. The figure shows that the most accurate algorithm is strongly connected to the active user and its neighbourhood. For example, the most accurate algorithm for the active User 0 is the *Absolute Original Rank Correlation* and the algorithm that produces the lowest MAE by setting User 100 as the active user is the *Absolute Cosine Co-Rated Similarity.*

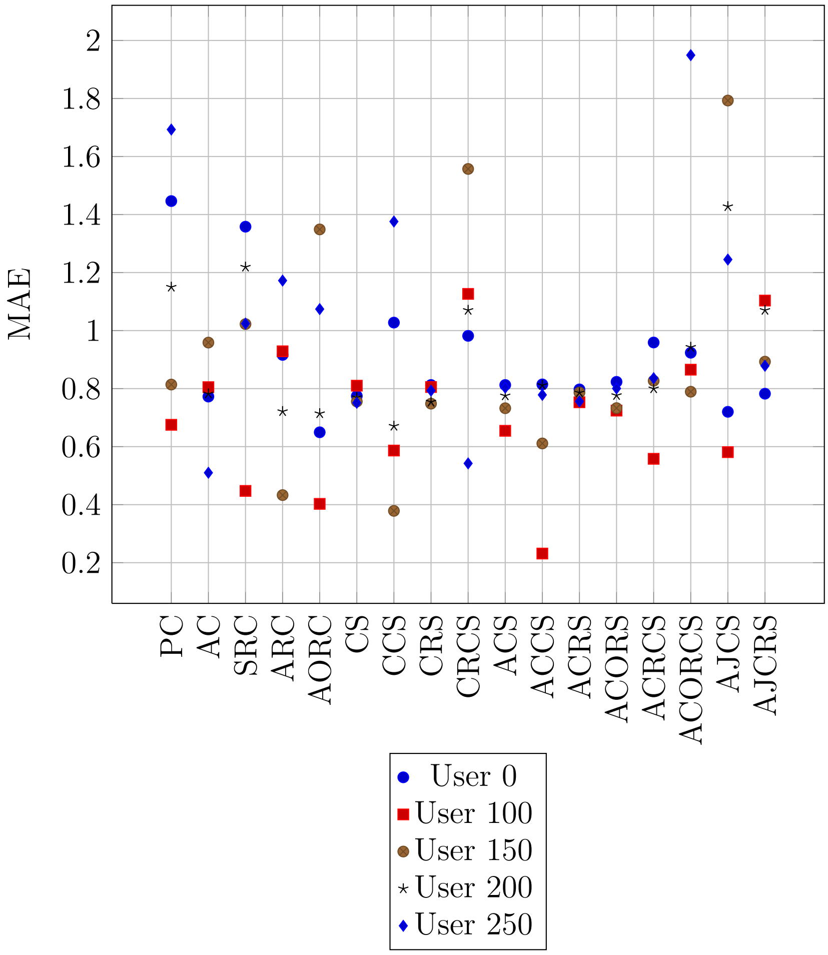


Figure 5.47: MovieLens - MAE - User-Based

5.4.2 Survey

5.4.2.1 Item-Based

The results that have been achieved by considering the item-based approach are presented in Figure 5.48. The test considers ten test cycles and the results are the average of the error rates. Figure 5.48 also presents the results by using different neighbour sizes. The neighbourhood also includes the active item.

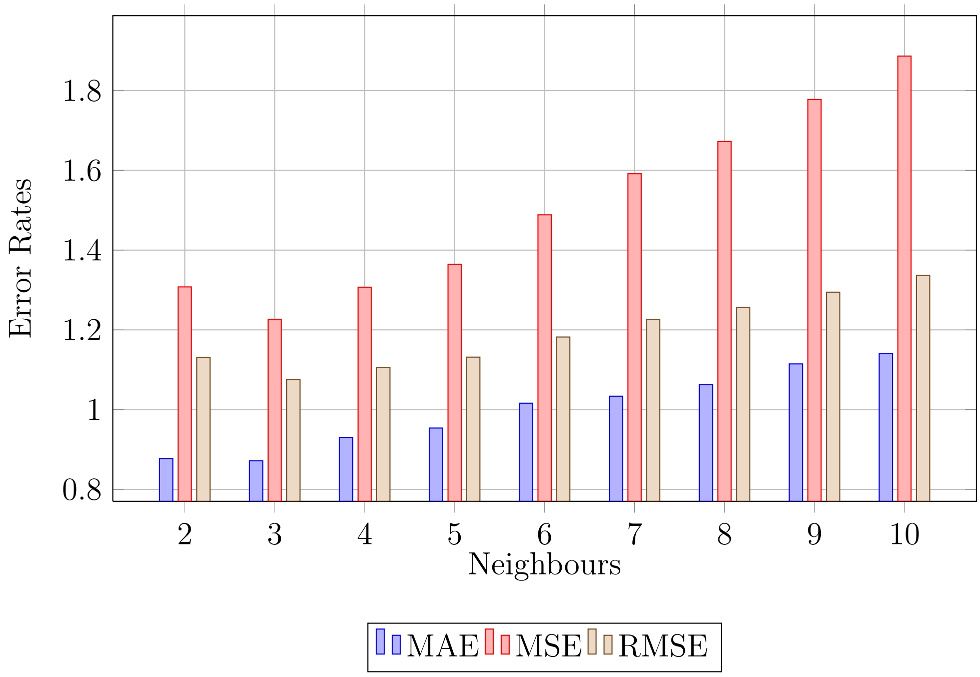


Figure 5.48: Error rates by considering the item-based approach, taking different neighbourhood sizes into account

5.4.2.2 User-Based

The results that have been achieved by considering the user-based approach are presented in Figure 5.49. The test considers ten test cycles and the results are the average of the error rates. Figure 5.49 also presents the results by using different neighbour sizes. The neighbourhood also includes the active user.

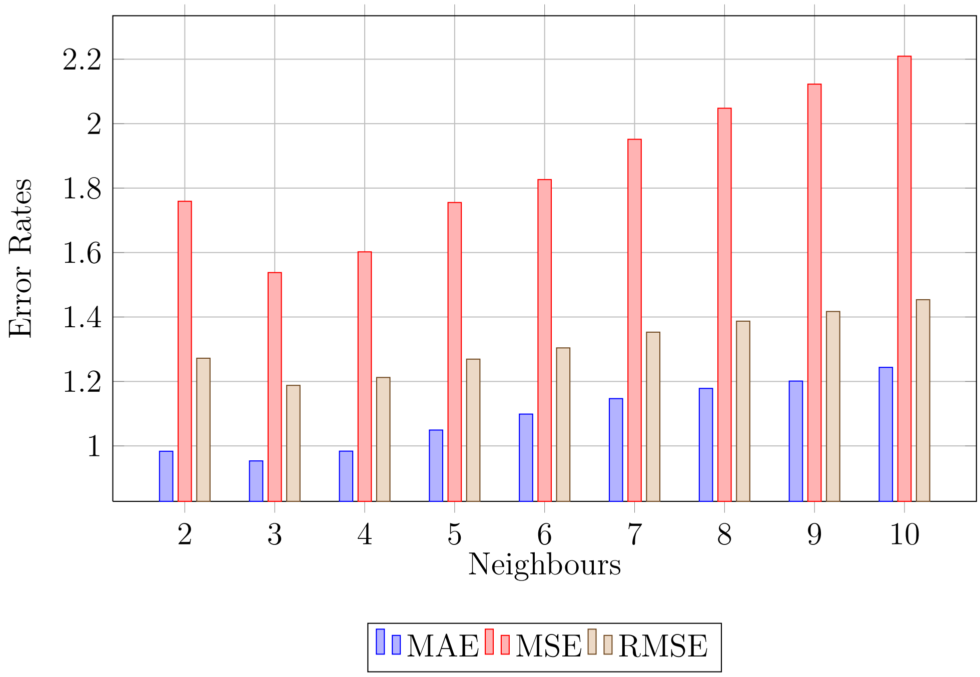


Figure 5.49: Error rates by considering the user-based approach, taking different neighbourhood sizes into account

5.4.2.3 Performance

The calculation duration of the tests is presented in Figure 5.50. The results are the average of ten test cycles. The figure presents the calculation duration of different neighbourhood sizes. The measurement unit is milliseconds (ms).

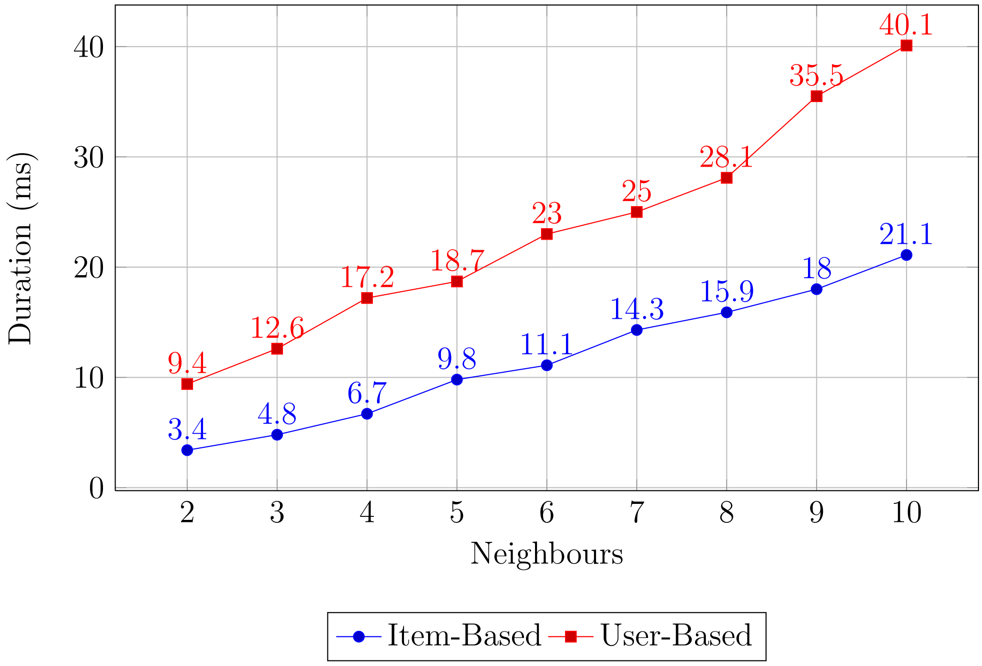


Figure 5.50: Calculation duration by using the dataset from the survey and taking different neighbourhood sizes into account

5.4.3 MovieLens

5.4.3.1 Item-Based

The results, which are presented in Figure 5.51, have been achieved by using the dataset from MovieLens. The presented averaged error rates are calculated by taking 10 test cycles into account. Additionally, the table presents the results that consider different neighbourhood sizes. 10, 20, 30, 40, and 50 k-nearest neighbours are considered. The calculation which produces these results uses the item-based approach.

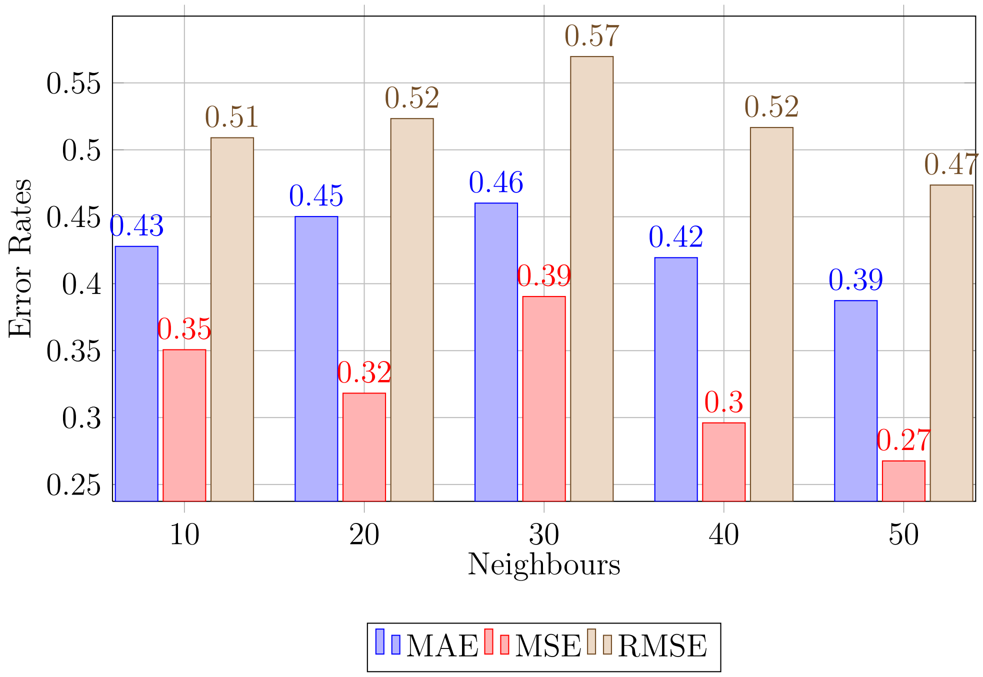


Figure 5.51: Error rates by considering the item-based approach, taking different neighbourhood sizes into account and using 10 test cycles

Figure 5.52 presents the error rates which have been achieved by averaging the errors by considering 50 test cycles.

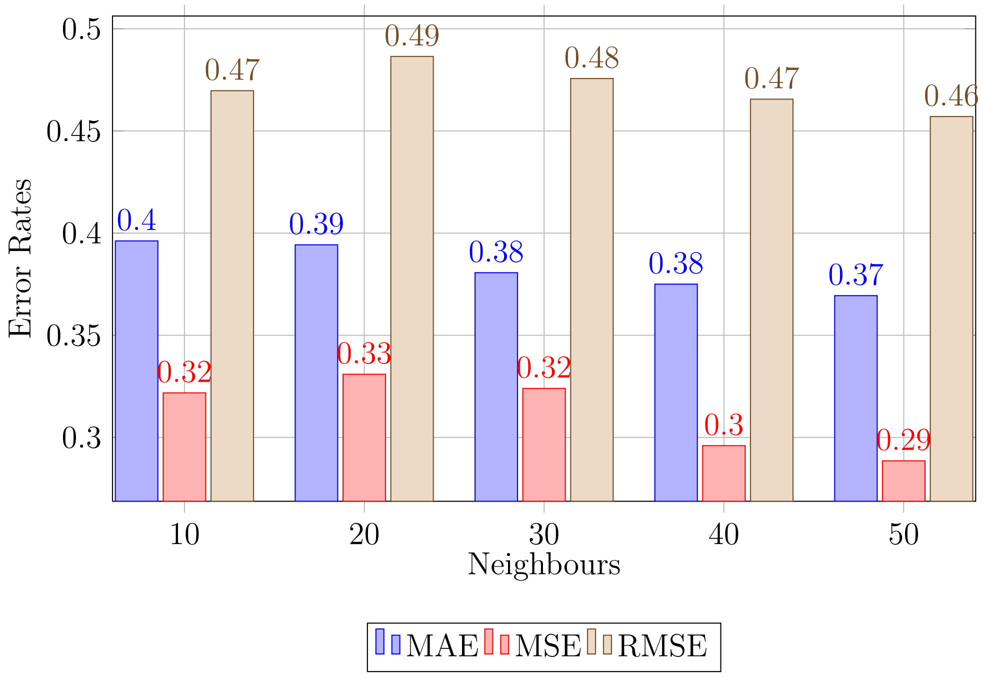


Figure 5.52: Error rates by considering the item-based approach, taking different neighbourhood sizes into account and using 50 test cycles

5.4.3.2 User-Based

The results which are presented by Figure 5.53 have been achieved by using the dataset from MovieLens. The presented averaged error rates are calculated by taking 10 test cycles into account. Additionally, the table presents the results that consider different neighbourhood sizes. 10, 20, 30, 40, and 50 k-nearest neighbours were considered. The calculation which produces these results uses the user-based approach.

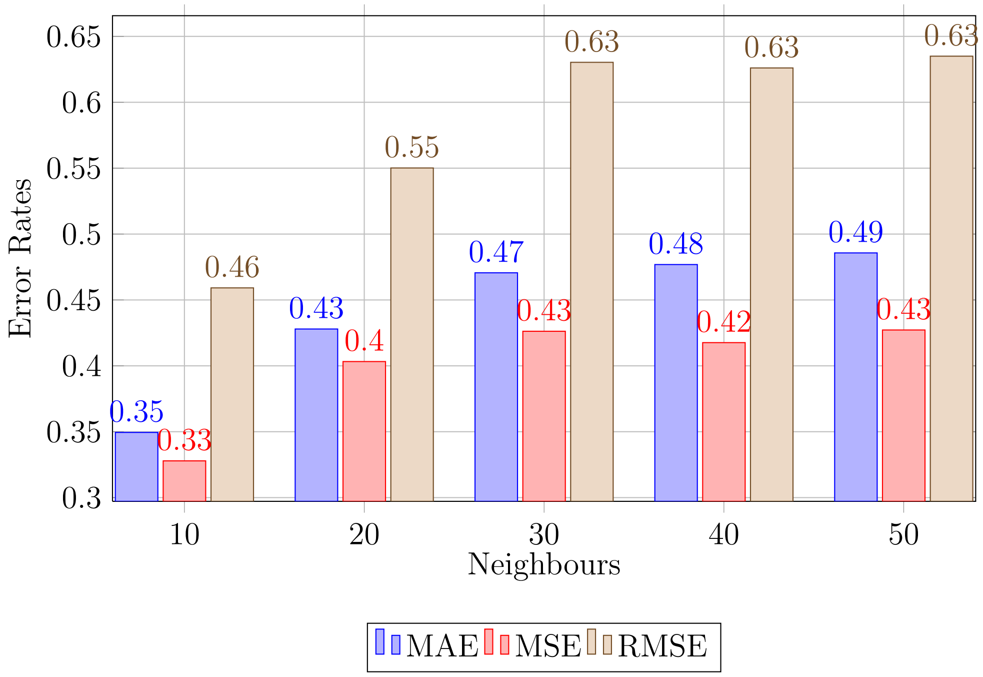


Figure 5.53: Error rates by considering the user-based approach, taking different neighbourhood sizes into account and using 10 test cycles

Figure 5.54 presents the error rates which have been achieved by averaging the errors by considering 50 test cycles.

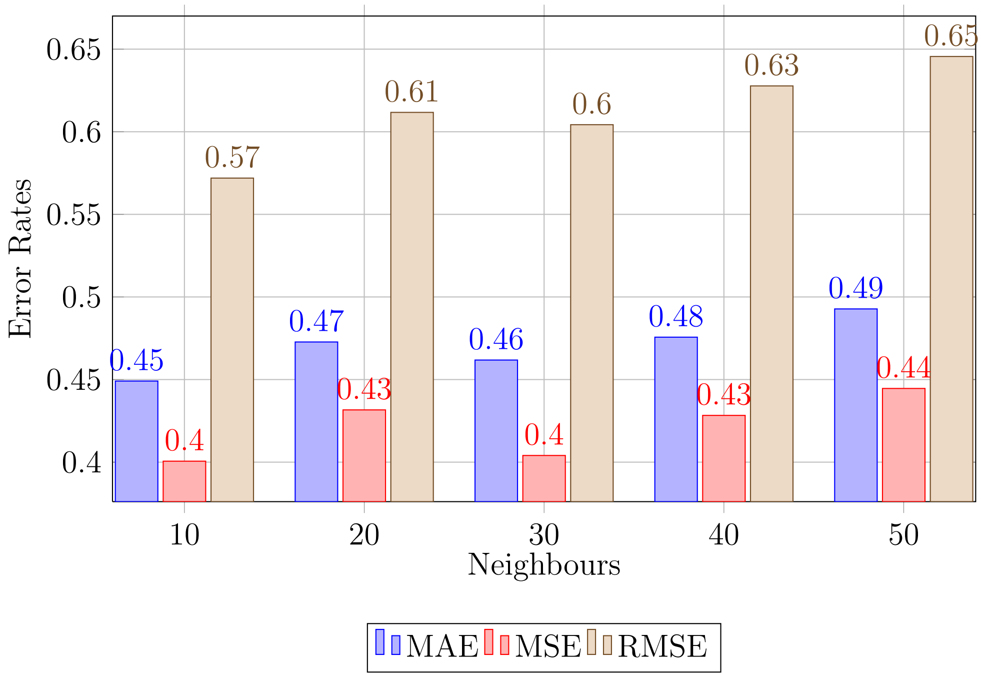


Figure 5.54: Error rates by considering the user-based approach, taking different neighbourhood sizes into account and using 50 test cycles

5.4.3.3 Performance

The following results were performed by using one Intel Core i7 with a CPU power of 2.7GHz and 8GB RAM. Figure 5.55 presents the results of this test. The milliseconds are the average by considering 50 test cycles, which been achieved by the error rate calculation. The results show that the increasing of the duration is quite linear. The duration of the calculations increases by ≈15 seconds if the number of neighbours increases by 10.

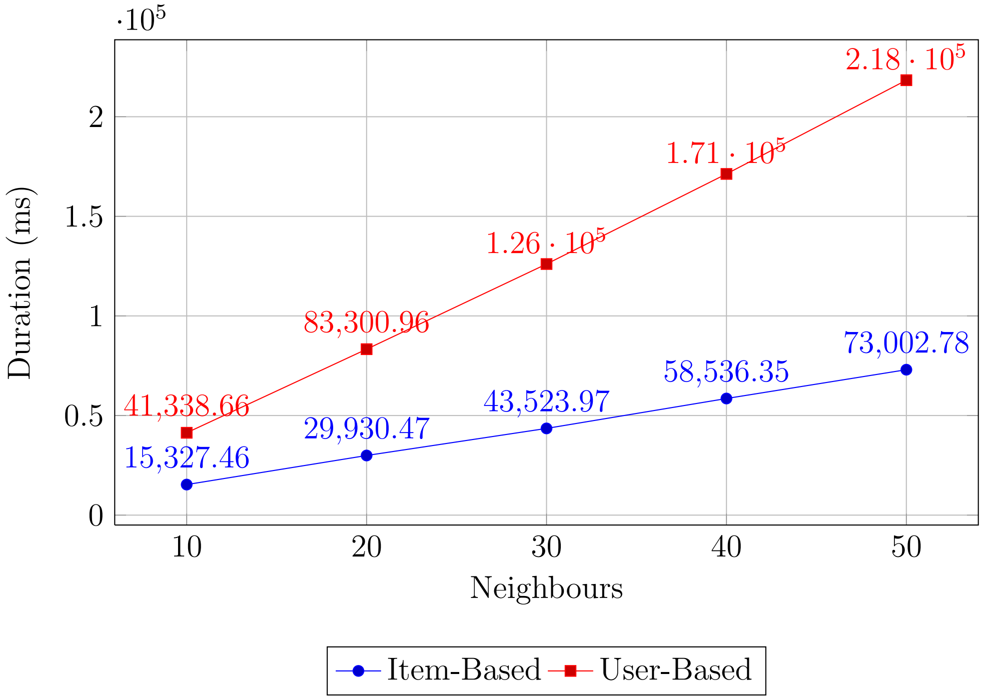


Figure 5.55: Calculation duration by using the dataset from MovieLens and taking different neighbourhood sizes into account

5.4.4 Conclusion

5.4.4.1 Error Rates

The results of the experiments of the proposed dynamic multi-algorithm collaborative-filtering system, which are presented in Section 5.4, prove the usefulness of the system. The results affirm the improvement of the predictions' accuracy. The error rates are significantly lower than the error rates from existing recommendation systems, which also use collaborative-filtering algorithms.

For example, Figure 5.48 and Figure 5.49 presented the error rates by using different neighbourhood sizes. These tables present the results that have been achieved by using the dataset from MovieLens. Figure 5.52 and Figure 5.54 presented the error rates by using the dataset from MovieLens. The results have been achieved by considering different neighbourhood sizes.

5.4.4.2 Performance

Section 5.4.3.3 presents the calculation durations that use the proposed dynamic multi-algorithm collaborative-filtering system. Figure 5.55 presents the duration of the calculation by considering the neighbourhood sizes of 10, 20, 30, 40, and 50 neighbours. In contrast to the calculation durations that use the k-nearest neighbour approach, the proposed dynamic system needs more time for the calculation. However, the experiments have been performed within one thread. A multi-threading could decrease the duration for the dynamic selection of the most accurate collaborative-filtering algorithm.

5.5 Comparison

This section presents the results of the comparison between the proposed dynamic multi-algorithm collaborative-filtering system and other existing recommendation systems. Figure 5.56 presents the mentioned comparison. Each of the compared systems use the dataset from MovieLens. In addition, the compared systems are evaluated by the usage of the MAE. The results prove the usefulness of the proposed system and prove the significant improvement of the predictions' accuracy.

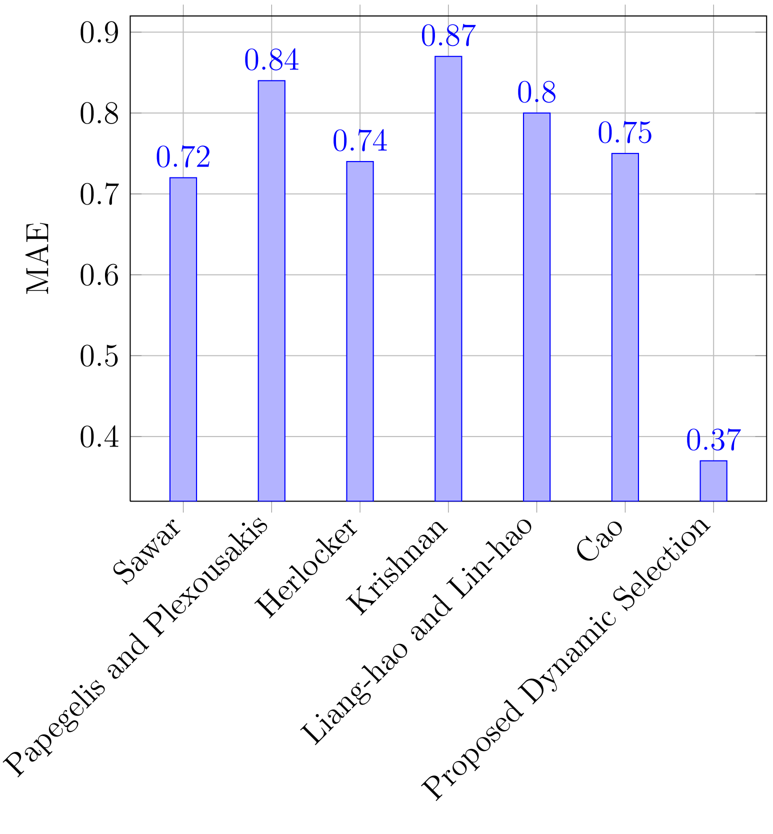


Figure 5.56: Comparison of the MAEs between existing recommendation systems and the proposed dynamic multi-algorithm collaborative-filtering system

5.6 Summary

Section 5 presented the results of the evaluation. The experiments of the evaluation considered two datasets.

The first dataset is a result from a survey. Respondents were asked to rate specified genres from an ETSI standard for Service Information. This standard specified twelve main genres. The respondents were able to rate these genres by setting values between 0 and 5, where 0 represents no interest in the selected genre and 5 definite interest in the selected genre. The results of the survey are presented in Table 3.1. This dataset represents a small community. The second dataset from MovieLens represents a large community. It contains ratings from 943 users and 1682 items (movies). It has been used by several other researchers to evaluate their systems.

The evaluation is focused on the accuracy of the predications. The predictions were calculated by using the *Weighted Sum* approach, which is defined by Equation 2.14. These calculated predictions were used to exploit the error rates, such as the MAE, the MSE, and the RMSE. The usage of these error rates offered the opportunity to compare the results from the proposed system with recommendation systems from other researchers.

In order to prove the usefulness of the proposed system, several experiments were undertaken. At the beginning a comparison between the results that use the truncation of the prediction and the results that do not use the truncation of the predictions was shown. The results prove that the predictions' truncation improve the predictions' accuracy compared to the classical approach, which does not use the truncation of predictions.

Besides the evaluation of the predictions' truncation, the usefulness of the proposed dynamic multi-algorithm collaborative-filtering system was proved by the errors' calculation that do not consider a neighbourhood, a errors' calculation that consider the neighbourhood, an a errors' calculation that takes the proposed dynamic approach into account.

The results prove that the usage of the neighbourhood improves the predications' accuracy compared to the approach that does not consider a neighbourhood. Additionally the results of the evaluation prove that the proposed dynamic selection is able to reduce the error rates significantly compared to existing approaches.

Chapter 6

Personal Program Guide - PPG

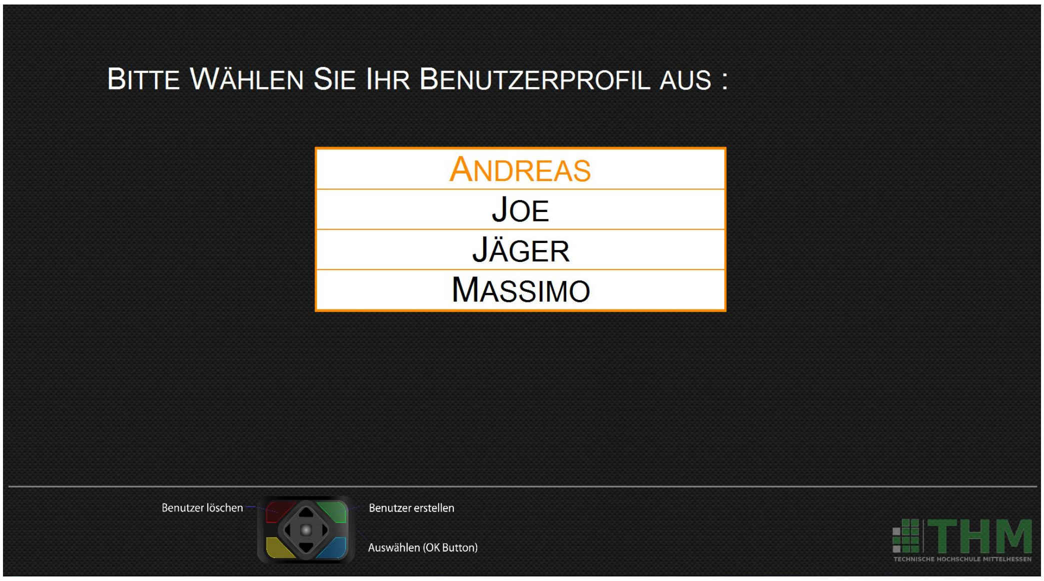
Since users are overloaded with information [27, 28], a personalized interface is needed which supports users in finding content of interest in less time [101]. Therefore an interface has been developed which is able to filter the immense amount of content and present personalized recommendations.

In contrast to a classical Electronic Program Guide (EPG) that does not provide personalized content recommendations [102], the PPG uses the developed and researched recommendation system which is described in Section 4. In addition it uses an API from Google. This API is able to access data from YouTube. The PPG uses this data and presents recommendations for events, which are broadcast from DVB and which are available from YouTube.

The following sections describe the features of the PPG and present the Graphical User Interface (GUI) of this interface.

6.1 Login

Users can use the login for the PPG which is shown by Figure 6.1. With the login, the PPG is able to identify the current user. The PPG will load the user profile, which can be created in an explicit and in an implicit manner. The user profiles are saved on the locally used device and/or on the File Transfer Protocol (FTP) [103] server, which is described in Section 3.1.



BITTE WÄHLEN SIE IHR BENUTZERPROFIL AUS :

ANDREAS
JOE
JÄGER
MASSIMO

Figure 6.1: PPG - Login

After the user has completed the login, the main menu of the PPG will be shown, as shown in Figure 6.2.

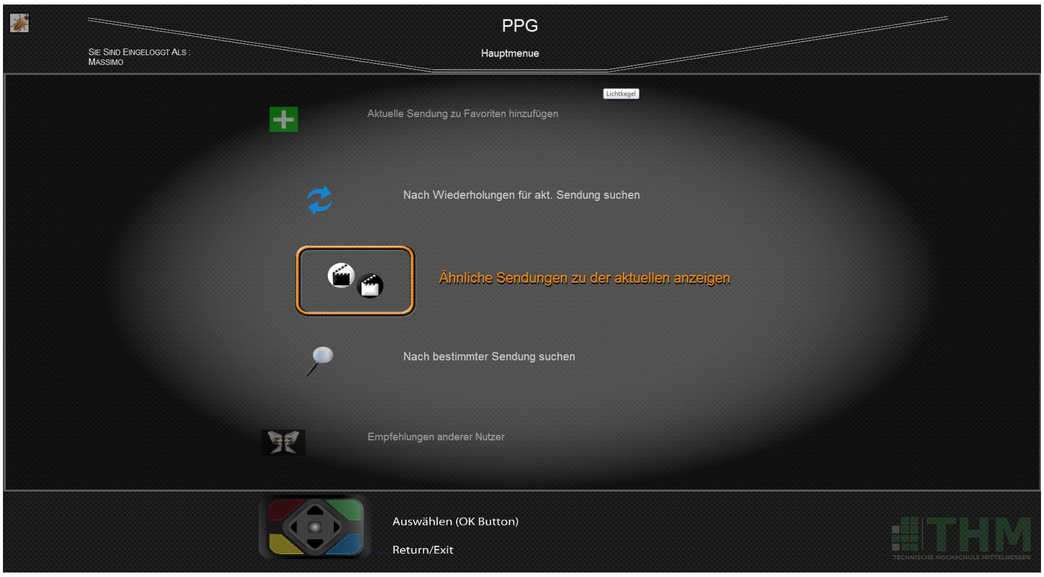
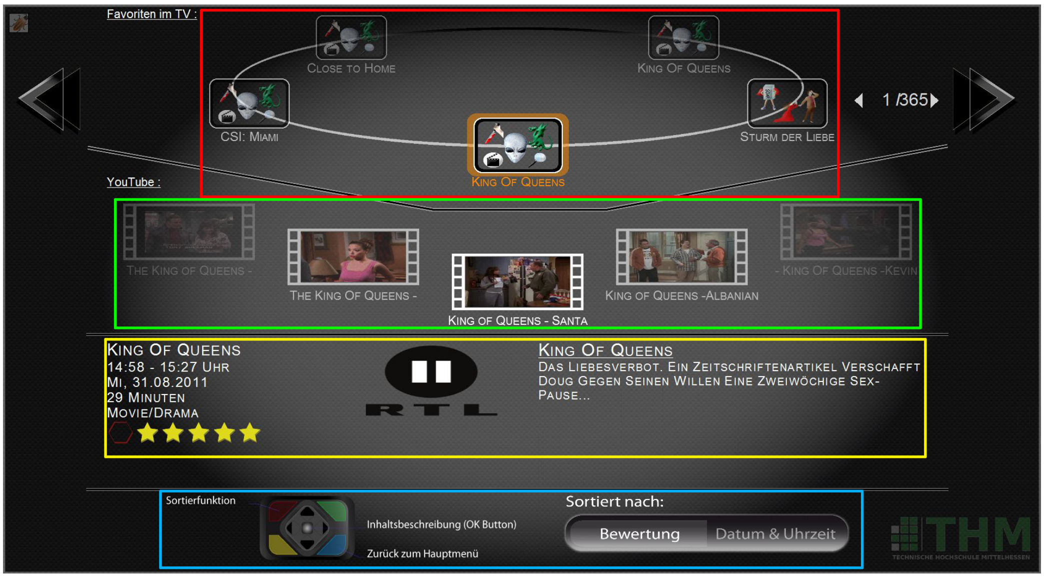

PPG
Sie Sind Eingeloggt Als :
Massimo
Hauptmenue
Lichtkegel
Aktuelle Sendung zu Favoriten hinzufügen
Nach Wiederholungen für akt. Sendung suchen
Ähnliche Sendungen zu der aktuellen anzeigen
Nach bestimmter Sendung suchen
Empfehlungen anderer Nutzer
Auswählen (OK Button)
Return/Exit
THM
TECHNISCHE HOCHSCHULE MITTELHESSEN

Figure 6.2: PPG - Main Menu

6.2 Viewing Content Related to the Current Event

The user is able to search content, which is related to the current broadcast event from DVB. Imagine that the current broadcast event is "King of Queens". The PPG will parse the available scheduled information from the EIT which is delivered by the DVB Transport Stream. Besides the events that are broadcast by DVB, the PPG also searches for video clips from YouTube that are related to the currently watched event. This is realized by the usage of the API from Google, which offers the opportunity to parse metadata from YoutTube. For example, the PPG sends a request to YouTube and asks for video clips with the title: "King of Queens". The PPG presents the related events and video clips, as shown in Figure 6.3.



King Of Queens
14:58 - 15:27 Uhr
Mi, 31.08.2011
29 Minuten
Movie/Drama
★★★★★

King Of Queens
Das Liebesverbot. Ein Zeitschriftenartikel verschafft Doug gegen seinen Willen eine zweiwöchige Sex-Pause...

Figure 6.3: PPG - Related Content

The figure shows that several features are realized. The windows is split into different areas. The area in the top (red rectangle) presents the events that are broadcast by DVB. The area in the middle (green rectangle) presents video clips from YouTube. The area in the bottom (yellow rectangle) presents the available metadata which describes the selected event or video clip in more detail. The window also includes a navigation bar (blue rectangle).

Figure 6.4 presents the process chart of this feature. The first step of this process includes the metadata extraction of the currently watched event. The used parser extracts the title, the genre, and the subgenre of this event. After the extraction is finished, the process parses the available scheduled information which contains metadata from the following events that will be broadcast in the near future. The process goes through the scheduled information and compares it with the currently watched event.

If the title, the genre, and the subgenre or the currently watched event is also part of the scheduled information, the relation between them is 100%. If an event's subgenre of the scheduled information is equal to the subgenre of the currently watched event, the relation between them is 80%. (If the subgenre is equal, the genre is equal, too, since a subgenre describes a genre in more detail). If a title and the genre of an event from the scheduled information is equal to the title and the genre to the currently watched event, the relation between them is 60%. If only the title of the current event is equal to an event of the scheduled information, the relationship is 40%. The relation is 20% if just the genre of an event from the scheduled information is equal to the currently watched event. If no parameter of an event from the scheduled information is equal to the currently watched event. the relation of

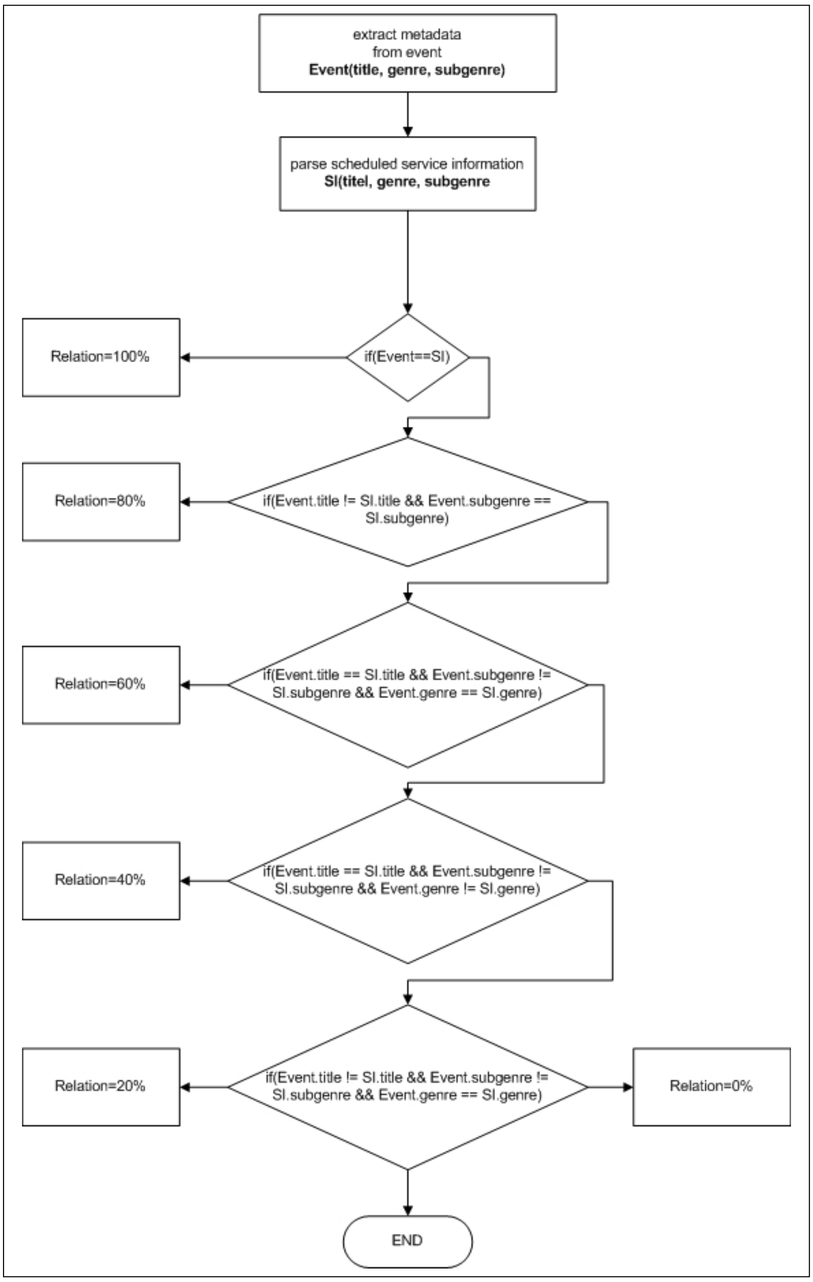

extract metadata
from event
Event(title, genre, subgenre)
parse scheduled service information
SI(titel, genre, subgenre
Relation=100%
if(Event==SI)
Relation=80%
if(Event.title != SI.title && Event.subgenre == SI.subgenre)
Relation=60%
if(Event.title == SI.title && Event.subgenre != SI.subgenre && Event.genre == SI.genre)
Relation=40%
if(Event.title == SI.title && Event.subgenre != SI.subgenre && Event.genre != SI.genre)
Relation=20%
if(Event.title != SI.title && Event.subgenre != SI.subgenre && Event.genre == SI.genre)
Relation=0%
END

Figure 6.4: PPG - Process Chart

scheduled event and the currently watchd event is 0%. The PPG uses these relations for the rating stars. 100% is presented with five stars, 80% with four stars, 60% with three stars, 40% with two stars, 20% with one star, and 0% is presented with zero stars.

6.3 Searching for Repeats of Current Event

The PPG is able to search for repeats of the currently watched event. The parser extracts the scheduled information which is broadcast by the DVB Transport Stream, and searches for events with the same title, subgenre, and genre. These events and video clips from YouTube, which will be delivered by Google's API by sending a request for video clips with the title of the currently watched event, will be presented to the user.

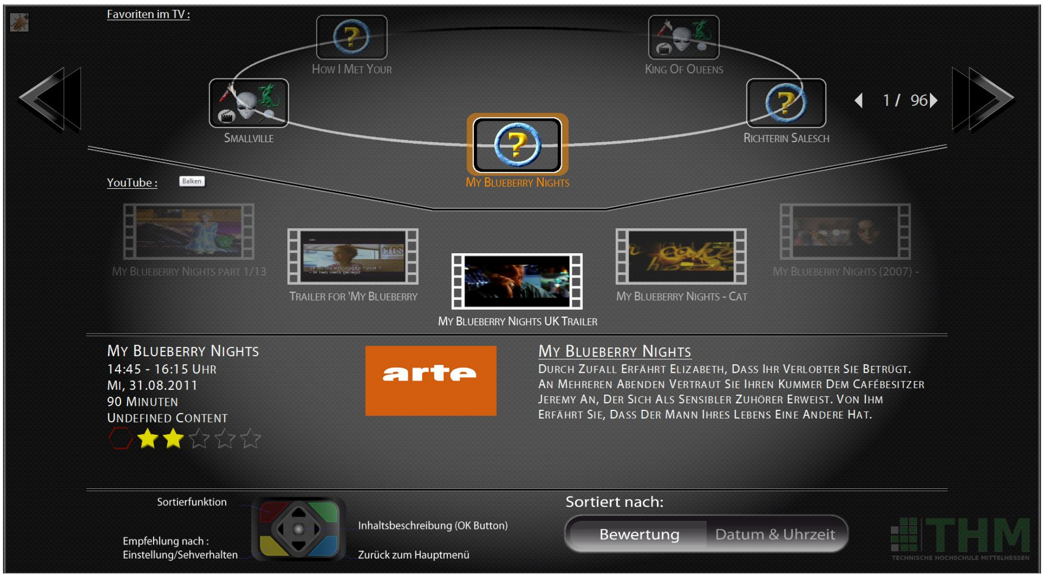
6.4 Adding Current Event to Favourites

With this menu entry, the user is able to easily add the currently watched event to her/his favourites. This event will be saved in the user profile of the currently logged-in user.

6.5 Viewing Recommendations

Figure 6.5 presents a screenshot of the PPG if the user wants to see the recommendations. The PPG extract the preferences of the currently logged-in user and parses the scheduled information for the creation of recommendations.

The recommendations are created by taking the equations from Section 4.2.1 into account. These equations consider the different available metadata from DVB, such as title of the event, genre, and subgenre of the event, and combine them for the creation of the RIs. The PPG will also search video clips from YouTube. At the beginning the PPG will send a request to YouTube, which searches for video clips with the title of the currently watched event. If the user switches to another event within the available DVB events, the PPG will send a new request to YouTube with the title of the newly selected event. Therefore the user will always get video clips from YouTube that are related to the currently selected event from the available recommendations.



My Blueberry Nights
14:45 - 16:15 Uhr
Mi, 31.08.2011
90 Minuten
Undefined Content
★★☆☆☆

My Blueberry Nights

Durch Zufall Erfährt Elizabeth, Dass Ihr Verlobter Sie Betrügt. An Mehreren Abenden Vertraut Sie Ihren Kummer Dem Cafébesitzer Jeremy An, Der Sich Als Sensibler Zuhörer Erweist. Von Ihm Erfährt Sie, Dass Der Mann Ihres Lebens Eine Andere Hat.

Figure 6.5: PPG - Recommendations

6.6 Viewing Recommendations for Today

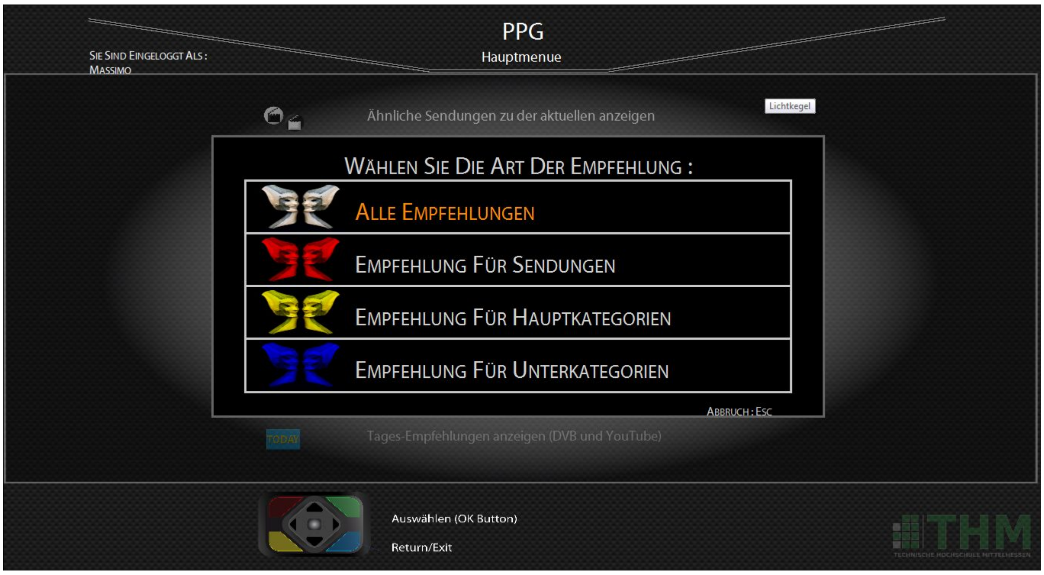
The presentation of the recommendations for today uses almost the same procedure as the presentation of recommendations which is described in Section 6.5. The difference between these two features is that the recommendations for today only searches events that will be broadcast on the same day.

6.7 Setting/Configuring Preferences

Since the user shall be able to set her/his preferences, the PPG offers the opportunity to create the explicit user profile. The user is able to rate genres and subgenres, which are specified by [20], by setting stars. Zero stars represent no interest in the selected genre/subgenre, five stars represent definite interest in the selected genre/subgenre. Besides these opportunities, users are also able to exclude genres and subgenres from the recommendations. If a user wants to exclude a genre/subgenre from the recommendations, she/he has to set the "stop sign". Figure 4.1 shows a screenshot of this feature. The set preferences will be saved in the explicit user profile. It will be saved in an XML file, which is presented in Listing 4.1.

6.8 Viewing of Collaborative Recommendations

The PPG is also able to recommend events, genres, and subgenres that are based on collaborative-filtering techniques. Figure 6.6 shows that the user is able to select whether all recommendations shall be presented, only recommendations that are based on the titles of events, or only recommendations that are based on genres or subgenres.



Sie Sind Eingeloggt Als: Massimo

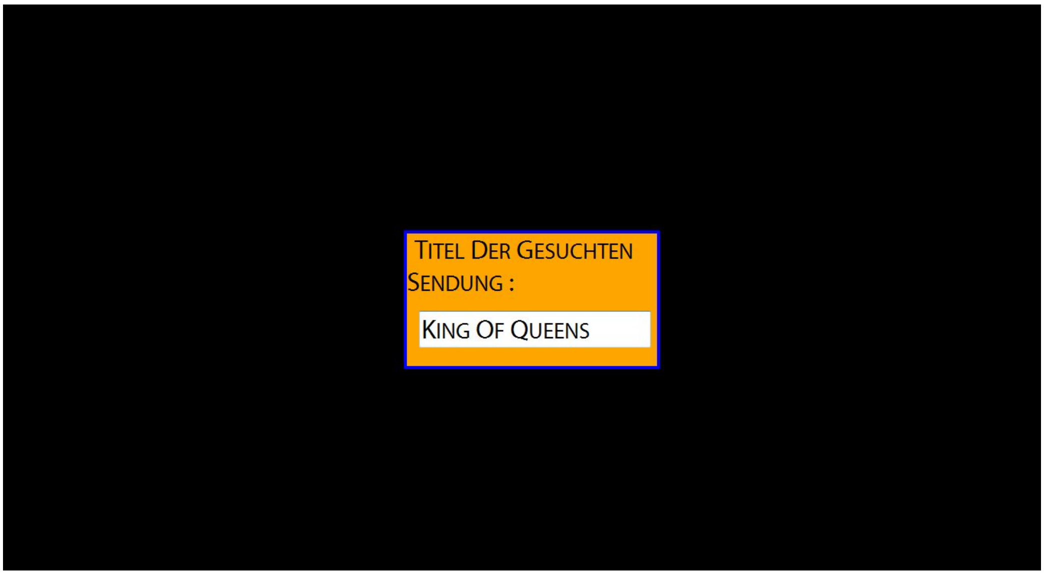
PPG
Hauptmenue
Ähnliche Sendungen zu der aktuellen anzeigen
Lichtkegel
Wählen Sie Die Art Der Empfehlung :
Alle Empfehlungen
Empfehlung Für Sendungen
Empfehlung Für Hauptkategorien
Empfehlung Für Unterkategorien
Abbruch : Esc
TODAY
Tages-Empfehlungen anzeigen (DVB und YouTube)
Auswählen (OK Button)
Return/Exit
THM

Figure 6.6: PPG - Collaborative Choice

If the user selects one of these choices, the PPG will present recommendations that are based on collaborative-filtering techniques. At the current stage, the PPG considers the user-based approach. It searches for events and genres/subgenres that have been watched by similar users. The PPG also searches for video clips from YouTube by using the API from Google.

6.9 Searching for an Event

The PPG also offers the opportunity to search for a specific event. Users are able to search for this event by putting the title of the requested event into a search bar, as shown in Figure 6.7. The PPG will parse the available scheduled information and present the event with the same title to the user. In addition the PPG presents video clips from YouTube. The PPG uses the API from Google and searches for video clips with the title that has been put into the search bar by the user.



Titel Der Gesuchten Sendung :

King Of Queens

Figure 6.7: PPG - Searching for an Event

6.10 Summary

This chapter presented the developed PPG. This PPG includes several features. These features use filtering techniques, such as content-based filtering and collaborative filtering. It also combines content which is broadcast by DVB and video clips from YouTube. The developed features can help users to find content of interest that is connected to their own preferences, which are created in an implicit or explicit manner. The recommendations can also be created according to the interests of similar users that have been archived by taking the collaborative-filtering approach into account.

Chapter 7

Conclusion and Future Work

This thesis tackles the collaborative-filtering topic by using a media environment. Since the quality of the predictions of a recommendation system is strongly connected to the accuracy of the predictions, the aim of this thesis focuses on the improvement of the predictions' accuracy. The system shall be able to consider small datasets, which contain just a small number of set ratings from users on items, e.g. movies. This small dataset shall represent a home environment, which can be found in families or blocks of flats. But the system shall not be limited to a small community. It shall also consider huge datasets, which represents a large community. Examples of huge datasets are MovieLens, Netflix, Amazon, and LastFM. This thesis presents a newly developed and researched dynamic multi-algorithm collaborative-filtering system, which is able to improve the prediction's accuracy significantly.

Although the presented thesis focuses on the prediction accuracy by using collaborative-filtering techniques, it also takes other topics into account.

Firstly the thesis introduces the reader to the building of user profiles.

The presented recommendation system considers the creation of user profiles that are built in an explicit and in an implicit manner. The explicit creation of user profiles is realized by the usage of an interface. Within this interface users are able to rate specified genres and subgenres which are broadcast by DVB. Besides these genres and subgenres, users are also able to rank events by titles. The implicit creation of the user profiles takes the viewing behaviour of individual users into account. The system logs the duration a user watches a genre, a subgenre, or an event. This logged duration is used to create an index, the RI, which represents the preference in the watched genre, subgenre, or event.

Secondly the thesis presents the two different approaches which are responsible for the filtering. On the one hand, the thesis describes the content-based approach. The content-based filtering approach uses the metadata which describes the content in more detail, for the creation of the recommendations. Since the focus of this thesis deals with the collaborative filtering techniques, the content-based approach is described only briefly. The collaborative filtering techniques, which build the main part of this thesis, are described in detail. The thesis presents several SotA collaborative filtering algorithms and shows possible weaknesses. Besides these SotA algorithms, the thesis also presents newly developed and researched collaborative-filtering algorithms which overcome the researched weaknesses. The thesis also describes the k-nearest neighbour approach, which is responsible for finding users or items that are quite similar to the active user or item. The predications' accuracy is calculated by the usage of the *Weighted Sum* approach, which is responsible for the predicting of single entries within the used user-item matrix. These predicted entries are compared with the “original” entry of the user-item matrix and error rates, such as MAE, MSE, or RMSE, give feedback on the accuracy of the used collaborative-filtering algorithm. The main contribution of this thesis is the dynamic selection of the most accurate collaborative filtering algorithm, which is strongly connected to the active user/item and its neighbourhood. This selection improves the prediction accuracy significantly compared to existing recommendation systems.

In addition these back-end topics, the thesis also presents an interface which is able to present the recommendations. This interface, the so-called PPG, considers the recommendations which are created by using the content-based and the collaborative filtering approach. It additionally takes DVB content and video clips from YouTube into account.

In order to prove the usefulness of the proposed system, the evaluation of this thesis considers two datasets. The first dataset represents the small community. The dataset was built by undertaking a survey. Respondents were asked to rate genres, which are specified by an ETSI Standard for Service Information. This standard specifies twelve main genres. The respondents were able to rate these genres by setting a value between 0 and 5, where 0 represents no interest in the selected genre and 5 definite interest in the selected genre. Twelve respondents took part in this survey. The results are saved within a user-item matrix and the experiments of the evaluation uses several variations of this user-item matrix. The second dataset contains ratings from 943 and 1682 items (movies). This dataset from MovieLens represents the huge community. The thesis uses this dataset because several existing systems are evaluated with this dataset. Therefore a comparison among existing systems and the proposed dynamic multi-algorithm collaborative-filtering system can be accomplished.

The experiments of the evaluation consider several aspects. At the beginning the evaluation shows the results which compare the error rates that have been achieved by using no truncation of predictions with the results of the error rates that use the truncation of the predictions. The results show that the predictions' truncation improves the predictions' accuracy compared to the classical approach, which does not use the truncation of predictions.

The usefulness of the proposed system is proved by comparing the error rates, which have been achieved by taking the following settings into account:

1. Without a neighbourhood
2. With a neighbourhood
3. With the proposed system

The experiments consider each of these settings and use the above mentioned datasets. The results show the error rates and the performance of these settings.

Without a Neighbourhood

The results of the experiments that do not use the k-nearest neighbour approach show that the most accurate filtering algorithm is strongly connected to the considered user-item matrix. This connection is proved by the results of the error rates by using the dataset from the survey. The results additionally prove the usefulness of the newly developed collaborative filtering algorithms. The results, which have been achieved by using the dataset from MovieLens, affirm the usefulness of this observation.

With a Neighbourhood

The experiments, which consider the k-nearest neighbours, prove that this approach is able to reduce the error rates compared to the experiments that do not use the k-nearest neighbours. The results also affirm the usefulness of the newly developed algorithms. These algorithms are able to produce a lower error rate than existing SotA collaborative filtering algorithms.

With the Proposed System

The evaluation of the proposed dynamic multi-algorithm collaborative-filtering system proves that this approach is able to reduce the error rates significantly, compared to the approaches that are described above. The results prove that the dynamic selection of the most accurate filtering algorithm by considering the k-nearest neighbours improves the predictions' accuracy and delivers an error rate that is significantly lower than existing approaches.

7.1 Future Work

In order to improve the performance of the proposed dynamic multi-algorithm collaborative-filtering system, the calculation of the error rates could be split. Currently the calculations are performed within one thread. The usage of multi-threading code can reduce the calculation duration. This could be useful if the proposed system shall be used in a real-time environment. In addition, the performance could be improved if the code was written in native code instead of C#. Since the evaluated system includes several collaborative-filtering algorithms, long-term tests could decrease the number of these algorithms. These long-term tests could be used to identify algorithms which could be deleted without a significant decline in the predictions' accuracy. These deletions could also improve the performance duration of the presented system.

Another aspect of future work could be the implementation in other environments. The proposed system is currently implemented in an entertainment environment. A next step could be the implementation into, for example, an online shop. Since some of the SotA algorithms are used in online shops, experiments could determine whether the proposed system can also be used in such an environment.

Companies in the Netherlands and in Austria have already asked for an integration of the recommendation system into their existing environment.

Bibliography

[1] Jong Seo Lee. Survey of recommender systems. *userscsccalpolyedu*, 2005.

[2] Loren Terveen and Will Hill. Beyond recommender systems: Helping people help each other. In *HCI in the New Millennium*, pages 487–509. Addison-Wesley, 2001.

[3] G. Adomavicius and A. Tuzhilin. Toward 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, 2005.

[4] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender Systems Handbook*, pages 1–35. 2011.

[5] J. Ben Schafer, Joseph Konstan, and John Riedi. Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce*, EC '99, pages 158–166, New York, NY, USA, 1999. ACM.

[6] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, 1992.

[7] J. Herlocker, J. Konstan, and J. Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, 5(4):287–310, 2002.

[8] Jianshu Weng, Chunyan Miao, Angela Goh, Zhiqi Shen, and Robert Gay. Trust-based agent community for collaborative recommendation. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, AAMAS '06, pages 1260–1262, New York, NY, USA, 2006. ACM.

[9] R. McLauglin and J. Herlocker. A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '04)*, pages 329–336, New York, NY, 2004. ACM.

[10] B. Sawar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web (WWW '01)*, pages 285–295, New York, NY, 2001. ACM.

[11] M. Papegelis and D. Plexousakis. Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents. *Engineering Applications of Artificial Intelligence*, 8:781–789, 2005.

[12] Dietmar Jannach and Gerhard Friedrich. Tutorial: Recommender systems. In *International Joint Conference on Artificial Intelligence in Barcelona*, 2011.

[13] P. Symeonidis, A. Nanopoulus, A. N. Papadopoulos, and Y. Manolopoulos. Collaborative filtering: Fallacies and insights in measuring similarity, 2006.

[14] Bin Cao, Qiang Yang, Jian-Tao Sun, and Zheng Chen. Learning bidirectional asymmetric similarity for collaborative filtering via matrix factorization. *Data Mining and Knowledge Discovery*, 22:393–418, 2011. 10.1007/s10618-011-0211-4.

[15] Vinod Krishnan, Pradeep K. Narayanashetty, Mukesh Nathan, Richard T. Davies, and Joseph A. Konstan. Who predicts better?: results from an online study comparing humans and an online recommender system. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 211–218, New York, NY, USA, 2008. ACM.

[16] Andreas Töscher, Michael Jahrer, and Robert M. Bell. The bigchaos solution to the netflix grand prize, 2009.

[17] Zheng Wen. Recommendation system based on collaborative filtering, 2008.

[18] Robert M. Bell and Yehuda Koren. Improved neighborhood-based collaborative filtering. 2007.

[19] E. Campochiaro, R. Casatta, P. Cremonesi, and R. Turrin. Do metrics make recommender algorithms? In *Advanced Information Networking and Applications Workshops, 2009. WAINA '09. International Conference on*, pages 648 –653, May 2009.

[20] European-Telecommunications-Standards-Institute. Specification for service information (si) in dvb systems. volume DVB Document A038, 2007.

[21] C. Überall, M. Rajarajan, V. Rakocevic, R. Jäger, and C. Köhnen. Recommendation index for dvb content using service information. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 1178–1181, New York, NY, 2009. IEEE.

[22] Christian Überall, Christopher Köhnen, Veselin Rakocevic, Rudolf Jäger, Erich Hoy, and Muttukrishnan Rajarajan. Recommendations in a heterogeneous service environment. *Multimedia Tools and Applications*, pages 1–36, 2011. 10.1007/s11042-011-0874-2.

[23] BITCOM. *Jeder zweite Fernseher ist internetfähig*, 2011 (accessed February 21, 2012). `http://www.bitkom.org/files/documents/BITKOM_PI_Smart_TV_25_03_2011.pdf`.

[24] European-Telecommunications-Standards-Institute. Digital video broadcasting (dvb), framing structure, channel coding and modulation for 11/12 ghz satellite services. Number V1.1.2, 1997.

[25] European-Telecommunications-Standards-Institute. Digital video broadcasting (dvb), framing structure, channel coding and modulation for cable systems. Number V1.2.1, 1998.

[26] European-Telecommunications-Standards-Institute. Digital video broadcasting (dvb), framing structure, channel coding and modulation for digital terrestrial television. Number V1.6.1, 2009.

[27] Paul Cotter and Barry Smyth. Ptv: Intelligent personalised tv guides. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 957–964. AAAI Press, 2000.

[28] Jiangshan Xu, Liang-Jie Zhang, Haiming Lu, and Yanda Li. The development and prospect of personalized tv program recommendation systems. In *Multimedia Software Engineering, 2002. Proceedings. Fourth International Symposium on*, pages 82 – 89, 2002.

[29] Hyoseop Shin, Minsoo Lee, and Eun Kim. Personalized digital tv content recommendation with integration of user behavior profiling and multimodal content rating. *IEEE Transactions on Consumer Electronics*, 55(3):1417 –1423, August 2009.

[30] Mauro Barbieri, Marco Ceccarelli, Gerhard Mekenkamp, and Jan Nesvadba. A personal tv receiver with storage and retrieval capabilities. In *UM'01 Workshop on Personalization in Future TV*, pages 13–14, 2001.

[31] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Improving the effectiveness of collaborative filtering on anonymous web usage data. In *Proceedings of the IJCAI 2001 Workshop on Intelligent Techniques for Web Personalization (ITWP '01)*, pages 53–60, 2001.

[32] X. Su and T. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009.

[33] J. Herlocker, J. Konstan, , A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '99)*, pages 230–237, New York, NY, 1999. ACM.

[34] M. I. Martín-Vicente, A. Gil-Solla, M. Ramos-Cabrer, Y. Blanco-Fernández, and M. López-Nores. Improving collaborative recommendation of coupons through digital tv by semantic inference of users' reputation. *IEEE Transactions on Consumer Electronics*, 57(1):178–186, 2011.

[35] W. Yang, Z. Wang, and M. You. An improved collaborative filtering method for recommendations' generation. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 4135–4139, The Hague, The Netherlands, 2004. IEEE.

[36] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4), 2002.

[37] S. Velusamy, L. Gopal, S. Bhatnagar, and S. Varadarajan. An efficient ad recommendation system for tv programs. *Multimedia Systems*, 14(2):73–87, 2008.

[38] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems*, 12(1):143–177, 2004.

[39] A. B. B. Martínez, J. j. P. Arias, A. F. Vilas, J. G. Duque, and M. L. Nores. What's on tv tonight? an efficient and effective personalized recommender system of tv programs. *IEEE Transactions on Consumer Electronics*, 55(1):286–294, 2009.

[40] Ji Liang-hao and Li Lin-hao. A new recommender model of collaborative filtering based on user. In *Management and Service Science (MASS), 2010 International Conference on*, pages 1 –5, August 2010.

[41] Jianping Fan, D.A. Keim, Yuli Gao, Hangzai Luo, and Zongmin Li. Justclick: Personalized image recommendation via exploratory search from large-scale flickr images. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(2):273 –288, feb. 2009.

[42] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76 – 80, jan/feb 2003.

[43] John Zimmerman, Kaushal Kurapati, Anna L. Buczak, Dave Schaffer, Srinivas Gutta, and Jacquelyn Martino. Chapter 5 tv personalization

system design of a tv show recommender engine and interface. In *Personalized Digital Television: Targeting Programs to Individual Viewers*, pages 27–51, 2004.

[44] Dagmar Kern, Michael Harding, Oliver Storz, Nigel Davis, and Albrecht Schmidt. Shaping how advertisers see me: User views on implicit and explicit profile capture. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 3363–3368, Florence, Italy, 2008. ACM.

[45] Toon De Pessemier and Luc Martens. A profile based recommendation system for tv-anytime annotated content. In *8th FirW PhD Symposium, Faculty of Engineering*, pages 104–105, Belgium, Gent, 2007.

[46] TV-Anytime. accessed February 28, 2012. `http://www.tv-anytime.org/`.

[47] Frank Hopfgartner and Joemon M. Jose. Semantic user profiling techniques for personalised multimedia recommendation. *Multimedia Systems*, 16(1):255–274, 2010.

[48] Iain Campbell and Keith Van Rijsbergen. The ostensive model of developing information needs. pages 251–268, 1996.

[49] Hongguang Zhang. Personalized tv program recommendation based on tv-anytime metadata. In *Proceedings of the Ninth International Symposium on Consumer Electronics 2005, ISCE 2005*, pages 242–246, China, Macou, 2005. IEEE.

[50] Ralf Klamma, Pham M. Cuong, and Yiwei Cao. You never walk alone: Recommending academic events based on social network analysis. In *Complex Sciences*, volume 4 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, chapter 64, pages 657–670. Springer, 2009.

[51] J. Nessel and B. Cimpa. The movieoracle - content based movie recommendations. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, volume 3, pages 361 –364, August 2011.

[52] C. Hayes, P. Massa, P. Avesani, and P. Cunningham. An on-line evaluation framework for recommender systems. In *In Workshop on Personalization and Recommendation in E-Commerce (Malaga*. Springer Verlag, 2002.

[53] Shang H. Hsu, Ming-Hui Wen, Hsin-Chieh Lin, Chun-Chia Lee, and Chia-Hoang Lee. Aimed: a personalized tv recommendation system. In *Proceedings of the 5th European conference on Interactive TV: a shared experience*, EuroITV'07, pages 166–174, Berlin, Heidelberg, 2007. Springer-Verlag.

[54] Chumki Basu, Haym Hirsh, and William Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *In Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 714–720. AAAI Press, 1998.

[55] Hilmi Yildirim and Mukkai S. Krishnamoorthy. A random walk method for alleviating the sparsity problem in collaborative filtering. In *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, pages 131–138, New York, NY, USA, 2008. ACM.

[56] Nathaniel Good, J. Ben Schafer, Joseph A. Konstan, Al Borchers, Badrul Sarwar, Jon Herlocker, and John Riedl. Combining collaborative filtering with personal agents for better recommendations. In *In Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pages 439–446, 1999.

[57] Yolanda Blanco-Fernández, José J. Pazos-Arias, Alberto Gil-Solla, Manuel Ramos-Cabrer, Martín López-Nores, and Belén Barragáns-Martínez. Avatar: Modeling users by dynamic ontologies in a tv recommender system based on semantic reasoning. In *Proceedings of the 3rd European Conference on Interactive TV*, 2005.

[58] Zan Huang, Hsinchun Chen, and Daniel Zeng. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems*, 22(1):116–142, 2004.

[59] Zan Huang, Daniel Zeng, and Hsinchun Chen. A link analysis approach to recommendation with sparse data. In *Americas conference on information systems*, AMCIS 2004, 2004.

[60] Danielle Hyunsook Lee. Pittcult: trust-based cultural event recommender. In *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, pages 311–314, New York, NY, USA, 2008. ACM.

[61] Jun Wang, Arjen P. de Vries, and Marcel J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 501–508, New York, NY, USA, 2006. ACM.

[62] John S. Breese, David Heckerman, and Carl Myers Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In Gregory F. Cooper and Serafín Moral, editors, *UAI*, pages 43–52. Morgan Kaufmann, 1998.

[63] Chumki Basu, Haym Hirsh, and William Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *In Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 714–720. AAAI Press, 1998.

[64] Lyle Ungar, Dean Foster, Ellen Andre, Star Wars, Fred Star Wars, Dean Star Wars, and Jason Hiver Whispers. Clustering methods for collaborative filtering. AAAI Press, 1998.

[65] Robert M. Bell and Yehuda Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pages 43–52, Washington, DC, USA, 2007. IEEE Computer Society.

[66] Toon De Pessemier, Sam Coppens, Kristof Geebelen, Chris Vleugels, Stijn Bannier, Erik Mannens, Kris Vanhecke, and Luc Martens. Collaborative recommendations with content-based filters for cultural activities via a scalable event distribution platform. *Multimedia Tools and Applications*, 58:167–213, 2012.

[67] Rong Hu and Pearl Pu. Enhancing collaborative filtering systems with personality information. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '11, pages 197–204, New York, NY, USA, 2011. ACM.

[68] Wikipedia. Pearson product-moment correlation coefficient, 2011.

[69] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53, 2004.

[70] MovieLens. Free, personalized, non-commercial movie recommendations, 2011.

[71] Y. Blanco-Fernandez, J. Pazos-arias, A. Gil-Solla, M. Ramos-Cabrer, and M. Lopez-Nores. Providing entertainment by content-based filtering and semantic reasoning in intelligent recommender systems. *Consumer Electronics, IEEE Transactions on*, 54(2):727 –735, May 2008.

[72] Liang Zhang, Deepak Agarwal, and Bee C. Chen. Generalizing matrix factorization through flexible regression priors. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '11, pages 13–20. ACM, 2011.

[73] Andreas Töscher, Michael Jahrer, and Robert Legenstein. Improved neighborhood-based algorithms for large-scale recommender systems. In *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, NETFLIX '08, pages 4:1–4:6, New York, NY, USA, 2008. ACM.

[74] Markus Zanker. A collaborative constraint-based meta-level recommender. In *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, pages 139–146, New York, NY, USA, 2008. ACM.

[75] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[76] G. Lekakos and P. Caravelas. A hybrid approach for movie recommendation. *Multimedia Tools and Applications*, 36(1):55–70, 2008.

[77] G. Karypis. Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the Tenth International Conference on Information and Knowledge Management, ACM CIKM 2001*, pages 247–254, Atlanta, Georgia, 2001. ACM.

[78] Alberto Gil-Solla Manuel Ramos-Cabrer Yolanda Blanco-Fernandez, Jose J. Pazos-Arias and Martin Lopez-Nores. A hybrid strategy to personalize the digital television by semantic inference. In *Interactive TV: A Shared Experience*, pages 33–51, 2008.

[79] Susan Gauch, Jason Chaffee, and Alexander Pretschner. Ontology-based personalized search and browsing. *Web Intelligence and Agent Systems*, 1:1–34, 2003.

[80] W3C. Web ontology language. 2007.

[81] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. The YouTube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 293–296, New York, NY, USA, 2010. ACM.

[82] L. Ardissono, C. Gena, P. Torasso, F. Bellifemine, A. Chiarotto, A. Difino, and B. Negro. Personalized recommendation of tv programs. In *Proceedings of the 8th AI*IA Conference*, 2003.

[83] W3C. Extensible markup language (xml). 2008.

[84] Robin Burke. Evaluating the dynamic properties of recommendation algorithms. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 225–228, New York, NY, USA, 2010. ACM.

[85] Iván Cantador, Alejandro Bellogín, and David Vallet. Content-based recommendation in social tagging systems. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 237–240, New York, NY, USA, 2010. ACM.

[86] M. Kayaalp, T. Ozyer, and S.T. Ozyer. A collaborative and content based event recommendation system integrated with data collection scrapers and services at a social networking site. In *International Conference on Advances in Social Network Analysis and Mining, 2009. ASONAM '09.*, pages 113 –118, July 2009.

[87] Zan Huang, D. Zeng, and Hsinchun Chen. A comparison of collaborative-filtering recommendation algorithms for e-commerce. *Intelligent Systems, IEEE*, 22(5):68 –78, September-October 2007.

[88] Bharath Kumar Mohan, Benjamin J. Keller, and Naren Ramakrishnan. Scouts, promoters, and connectors: The roles of ratings in nearest-neighbor collaborative filtering. *ACM Transactions on the Web*, 1(2), 2007.

[89] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76 – 80, January/February 2003.

[90] Manos Papagelis, Dimitris Plexousakis, and Themistoklis Kutsuras. Alleviating the sparsity problem of collaborative filtering using trust inferences. In *Proceedings of the Third international conference on Trust Management*, iTrust'05, pages 224–239, Berlin, Heidelberg, 2005. Springer-Verlag.

[91] Markus Zanker and Markus Jessenitschnig. Case-studies on exploiting explicit customer requirements in recommender systems. In *User modeling and user-adapted interaction: The journal of personalization research, A. Tuzhilin and B. Mobasher (eds.): Special issue on Data mining and personalization*, pages 133–166, 2009.

[92] Mohak Sharma and P. Krishna Reddy. Using lower-bound similarity to enhance the performance of recommender systems. In *Proceedings of*

the Fourth Annual ACM Bangalore Conference, COMPUTE '11, pages 22:1–22:4, New York, NY, USA, 2011. ACM.

[93] T. Isobe, M. Fujiwara, H. Kaneta, N. Uratani, and T. Morita. Development and features of a tv navigation system. *IEEE Transactions on Consumer Electronics*, 49(4):1035 – 1042, November 2003.

[94] International Organization for Standardization. Information technology - generic coding of moving pictures and associated audio information: Systems, 2000.

[95] David M. Nichols. Implicit rating and filtering. In *In Proceedings of the Fifth DELOS Workshop on Filtering and Collaborative Filtering*, pages 31–36, 1998.

[96] Kaushal Kurapati Srinivas, Srinivas Gutta, David Schaffer, Jacquelyn Martino, and John Zimmerman. A multi-agent tv recommender. In *In Proceedings of the UM 2001 workshop Personalization in Future TV*, 2001.

[97] Micheline Hancock-Beaulieu and Stephen Walker. An evaluation of automatic query expansion in an online library catalogue. *J. Doc.*, 48:406–421, December 1992.

[98] A. Spink, H. Greisdorf, and J. Bateman. From highly relevant to not relevant: examining different regions of relevance. *Information Processing Manage*, 34(5):599 –621, 1998.

[99] Stefania Costache, Julien Gaugaz, Ekaterini Ioannou, Claudia Niederée, and Wolfgang Nejdl. Detecting contexts on the desktop using bayesian networks. In *ACM SIGIR Conference on Research and Developmenting Information Retrieval, SIGIR*, pages 3–6, Switzerland, Geneva, 2010. ACM.

[100] L. Papula. Mathematische formelsammlung. volume 8. vieweg Verlag, 2003.

[101] L. Ardissono, F. Portis, P. Torasso, F. Bellifemine, A. Chiarotto, and A. Difino. Architecture of a system for the generation of personalized electronic program guides. In *Proc. UM'01 Workshop on Personalization in Future TV*, 2001.

[102] Zhiwen Yu and Xingshe Zhou. Tv3p: an adaptive assistant for personalized tv. *IEEE Transactions on Consumer Electronics*, 50(1):393 – 399, February 2004.

[103] RFC 959. File transfer protocol. 1985.

[104] Christian Überall, Christopher Köhnen, Rudolf Jäger, Veselin Rakocevic, and Muttukrishnan Rajarajan. A dynamic multi-algorithm collaborative-filtering system. *Data Mining and Knowledge Discovery*, 2012. under review.

Publications

Journals

[1] Christian Überall, Christopher Köhnen, Veselin Rakocevic, Rudolf Jäger, Erich Hoy, and Muttukrishnan Rajarajan. Recommendations in a heterogeneous service environment. *Multimedia Tools and Applications*, pages 1–36, 2011. 10.1007/s11042-011-0874-2.

[2] Christian Überall, Christopher Köhnen, Rudolf Jäger, Veselin Rakocevic, Muttukrishnan Rajarajan, A dynamic multi-algorithm collaborative-filtering system, *Data Mining and Knowledge Discovery*, 2012, under review

Conferences

[3] R. Jäger, C. Köhnen, C. Überall, M. Becker, F. Bellot, System Architecture for advanced iTV Services in an IPTV Environment, *International Conference on Telecommunication and Multimedia*, Ierapetra (Crete), Greece, July 16-18 2008, 2008

[4] Christian Überall, Muttukrishnan Rajarajan, Veselin Rakocevic, Rudolf Jäger, Christopher Köhnen, Recommendation Index for DVB content using Service Information, *Proceeding of the ICME*, IEEE, 2009

[5] C. Köhnen, C. Überall, V. Rakocevic, M. Rajarajan, R. Jäger, QoSi-LAN - A Heterogeneous Approach to Quality of Service in Local Area Networks, *2010 Second International Conferences on Advances in Multimedia (MMEDIA)*, vol., no., pp.109-112, 13-19 June 2010

[6] C. Köhnen, C. Überall, F. Adamsky, V. Rakocevic, M. Rajarajan, R. Jäger, Enhancements to Statistical Protocol IDentification (SPID) for Self-Organised QoS in LANs, *19. International Conference on Computer Communication Networks*, 2-5 August 2010

[7] Florian Adamsky, Christopher Köhnen, Christian Überall, Veselin Rakocevic, Muttukrishnan Rajarajan, Rudolf Jäger, A Novel Concept for Hybrid Quality Improvements in Consumer Networks, *The 1st IEEE International Conference on Consumer Electronics*, 2011

EU Project

Author

[8] Oskar van Deventer, Mark Gülbahar, Sebastian Schumann, Radovan Kadlic, Gregor Rozinaj, Ivan Minarik, Joost de Wit, Christian Überall, Christian Köbel. ANALYSIS: Multi-User, Multimodal & Context Aware Value Added Services. *EU Project Hbb-Next Deliverable*, 2012

Contribution

[9] Sven Glaser, Bettina Heidkamp, Jennifer Müller, Jeroen Vanattenhoven, David Geerts, Sachin Agarwal, Janina Renz, Christian Überall, Oskar van Deventer, Ray van Brandenburg. Usage Scenarios and Use Cases. *EU Project Hbb-Next Deliverable*, 2011