Multi-dimensional clustering in user profiling

Ayşe Çufoğlu

School of Electronics and Computer Science

MULTI-DIMENSIONAL CLUSTERING IN USER PROFILING

Ayşe Çufoğlu

A thesis submitted in partial fulfilment of the requirements of the

University of Westminster for the degree of

Doctor of Philosophy

April 2012

Dedicated to my parents

Arzu and Veli Çufoğlu

I declare that the work presented in this thesis is my own, has not been submitted for any other award, is identical to the content of the electronic submission and that, to the best of my knowledge, it does not contain any material created by another person, except where due reference is made.

Ayşe Çufoğlu

Acknowledgment

In the first place I would like to express my sincere gratitude to Dr. Mahi Lohi and Prof. Kambiz Madani for their guidance, supervision and advice from the very early stage of this research.

I also owe my deepest gratitude to Mr. Colin Everiss for his support and contribution.

My special thanks to Adem Coskun for his excellent friendship and continuous support.

I am deeply grateful to have such a supportive and caring family. I would like to express my deepest gratitude to my parents Arzu and Veli Çufoğlu, my twin brothers Kemal and Ismail, my sister Fatma, my brother-in-law Alihan and my cousin Rengin Hudaverdi.

I would like to thank to all my colleagues for their help, support and excellent friendship.

Finally, I would like to thank everybody who was important to the successful realization of this project, as well as expressing my apology that I could not mention them personally one by one.

Abstract

User profiling has attracted an enormous number of technological methods and applications. With the increasing amount of products and services, user profiling has created opportunities to catch the attention of the user as well as achieving high user satisfaction. To provide the user what she/he wants, when and how, depends largely on understanding them. The user profile is the representation of the user and holds the information about the user. These profiles are the outcome of the user profiling.

Personalization is the adaptation of the services to meet the user's needs and expectations. Therefore, the knowledge about the user leads to a personalized user experience. In user profiling applications the major challenge is to build and handle user profiles. In the literature there are two main user profiling methods, collaborative and the content-based. Apart from these traditional profiling methods, a number of classification and clustering algorithms have been used to classify user related information to create user profiles. However, the profiling, achieved through these works, is lacking in terms of accuracy. This is because, all information within the profile has the same influence during the profiling even though some are irrelevant user information.

In this thesis, a primary aim is to provide an insight into the concept of user profiling. For this purpose a comprehensive background study of the literature was conducted and summarized in this thesis. Furthermore, existing user profiling methods as well as the classification and clustering algorithms were

investigated. Being one of the objectives of this study, the use of these algorithms for user profiling was examined. A number of classification and clustering algorithms, such as Bayesian Networks (BN) and Decision Trees (DTs) have been simulated using user profiles and their classification accuracy performances were evaluated. Additionally, a novel clustering algorithm for the user profiling, namely Multi-Dimensional Clustering (MDC), has been proposed.

The MDC is a modified version of the Instance Based Learner (IBL) algorithm. In IBL every feature has an equal effect on the classification regardless of their relevance. MDC differs from the IBL by assigning weights to feature values to distinguish the effect of the features on clustering. Existing feature weighing methods, for instance Cross Category Feature (CCF), has also been investigated. In this thesis, three feature value weighting methods have been proposed for the MDC. These methods are; MDC weight method by Cross Clustering (MDC-CC), MDC weight method by Balanced Clustering (MDC-BC) and MDC weight method by changing the Lower-limit to Zero (MDC-LZ). All of these weighted MDC algorithms have been tested and evaluated. Additional simulations were carried out with existing weighted and non-weighted IBL algorithms (i.e. K-Star and Locally Weighted Learning (LWL)) in order to demonstrate the performance of the proposed methods. Furthermore, a real life scenario is implemented to show how the MDC can be used for the user profiling to improve personalized service provisioning in mobile environments.

The experiments presented in this thesis were conducted by using user profile datasets that reflect the user's personal information, preferences and interests. The simulations with existing classification and clustering algorithms (e.g.

Bayesian Networks (BN), Naïve Bayesian (NB), Lazy learning of Bayesian Rules (LBR), Iterative Dichotomister 3 (Id3)) were performed on the WEKA (version 3.5.7) machine learning platform. WEKA serves as a workbench to work with a collection of popular learning schemes implemented in JAVA. In addition, the MDC-CC, MDC-BC and MDC-LZ have been implemented on NetBeans IDE 6.1 Beta as a JAVA application and MATLAB. Finally, the real life scenario is implemented as a Java Mobile Application (Java ME) on NetBeans IDE 7.1. All simulation results were evaluated based on the error rate and accuracy.

Table of Contents

List of Figures

Chapter 2

Chapter 3

Chapter 4

Chapter 5

List of Tables

Chapter 2

Chapter 3

Chapter 5

Abbreviations

3GPP	*3rd Generation Partnership Project*
AI	*Artificial Intelligence*
API	*Application Programming Interface*
AUC	*Area Under Curve*
BC	*Balanced Clustering*
BN	*Bayesian Network*
CART	*Classification and Regression Tree*
CC	*Cross Clustering*
CCF	*Cross-Category Feature*
CC/PP	*Composite Capability/ Preference Profiles*
CF	*Collaborative Filtering*
DBSCAN	*Density Based Spatial Clustering of Applications with Noise*
DM	*Data Mining*
DTs	*Decision Trees*
DTV	*Digital Television*
ETSI	*European Telecommunications Standards Institute*
GPS	*Global Positioning System*

GUP	*Generic User Profile*
IBL	*Instance Based Learner*
IDF	*Inverse Document Frequency*
iDTV	*Integrated Digital Television*
ID3	*Iterative Dichotomister 3*
LBR	*Lazy learning of Bayesian Rules*
LIA	*Learning Information Agents*
LNB	*Lazy Naïve Bayesian*
LSI	*Latent Semantic Indexing*
LWL	*Locally Weighted Learning*
LWNB	*Locally Weighted Naïve Bayesian*
LZ	*Lower-limit to Zero*
MAE	*Mean Absolute Error*
MCDA	*Multi Criteria Decision Analysis*
MDC	*Multi-Dimensional Clustering*
MI	*Mutual Information*
ML	*Machine Learning*
MSD	*Mean Square Difference*
NB	*Naïve Bayesian*

NBTree	*Naïve Bayesian Tree*
NN	*Nearest Neighbour*
NNs	*Neural Networks*
NNA	*Neural Network Agents*
OMA	*Open Mobile Alliance*
PCC	*Pearson Correlation Coefficient*
PCF	*Per-Category Feature*
PDA	*Personal Digital Assistant*
RAE	*Relative Absolute Error*
RMSE	*Root Mean Squared Error*
RRSE	*Root Relative Squared Error*
SMO	*Sequential Minimal Optimization*
SNNB	*Selective Neighbourhood based Naïve Bayesian*
SVD	*Singular Value Decomposition*
SVM	*Support Vector Machine*
SVs	*Support Vectors*
TAN	*Tree Augmented Naïve Bayesian*
TF	*Term Frequency*

UAProf	*Universal Agent Profile*
UCI	*University of California, Irvine*
VDM	*Value Difference Matrix*
VSM	*Vector Space Model*
WAP	*Wireless Application Protocol*
WEKA	*Weikato Environment for Knowledge Analysis*
W3C	*World Wide Web Consortium*

Chapter 1

Introduction

Today, we are living in a communication era where numerous services are available for the customers across many devices (i.e. web, mobile, tablet). In a competitive market therefore, user profiles have become very important for service providers to attract user's attention and get noticed among others. User profiles make service personalization possible, which improves quality of service and optimizes the user satisfaction.

Personalized services aim to match users' requirements by considering when, where and how the users require the service to be delivered. The success of these applications relies on how well the service provider knows the user requirements and how well this can be reflected on the services. The description of the user interests, preferences, characteristics and needs are defined as user profiles [1]-[4]. The practice of gathering, organizing and interpreting the user profile information is called user profiling [5][6]. User profiles include a variety of information about each user such as personal profile data (demographic profile data), interest profile data and preference profile data.

The main challenge in personalization applications is the user profile initialization for the new user and the continuous updating of the existing user's profile information based on the user's changing needs, interests and preferences. In literature there are two main user profiling methods, collaborative and content-based. Collaborative method assumes that the users, who belong to the same group (e.g. age, sex, social class) behave similarly, and therefore have similar profiles [1]. Content-based method, on the other hand, assumes that the users show the same particular behaviour under the same circumstances [1].

Various works can be found in the literature for collaborative and content-based user profiling [7]-[11]. However, user profiling methods have limitations when compared to each other. For instance, the collaborative method, suffers from 'sparsity' and 'new user' problems. The 'sparsity' is the poor prediction capabilities of new item due to lack of ratings on the item [12]. The 'new user' problem, on the other hand, is when poor recommendations are made to the new users due to the lack of ratings in their profiles [12]. The 'synonym' and 'polysemy' are the limitations of the content-based method caused by its content dependence characteristic. In content-based method it is also hard to introduce serendipitous recommendations as only user's previous feedbacks considered for the future recommendations. In the literature, hybrid user profiling has been proposed to overcome the aforementioned limitations by combining the methods. However, user profiles that are created based on the above mentioned user profiling methods are not adequate to personalize different services. This project aimed to focus on this problem and propose the most efficient algorithm for user profiling where user profile data of a single service

(i.e. music recommendation) can be used successfully with other services (i.e. restaurant recommendation).

For this purpose, the objective of this research program is to investigate the existing user profiling methods, clustering and classification algorithms and the feature weighting methods and propose a new weighted clustering algorithm for the user profiling. The research methodology is explained in the next section. Section 1.2 presents the main contributions of this research. The outline of the rest of this thesis is given in Section 1.3.

1.1. Research Aims, Objectives and Methodology

The aims and objectives of the thesis can be listed as follows:

1. Investigating the existing user profiling methods and classification and clustering algorithms for the user profiling.
2. Investigating the existing feature weighting methods for the user profiling.
3. To propose and implement a novel weighted clustering algorithm using a combination of classification and clustering algorithms for the purpose of improving the accuracy of existing methods of user profiling.

1.2. Contributions

The following are the main contributions and the related publications resulting from this research program;

- This work investigated the classification accuracy performance of the NB, IB1, BN and LBR classifiers on the user profile. The results of this study

were published in IEEE International Conference on Computer Engineering and Systems (ICCES'08).

- This work compared the classification accuracy in user profiling. Performance of the classifiers was published in IEEE Seventh International Conference on Machine Learning and Applications (ICMLA'08).

- This work investigated 11 well known classifiers and compared their classification accuracy on 4 different user profiles. The results of this study were published in IEEE World Congress on Computer Science and Information Engineering (CSIE'09).

- This work proposed a weighted classification method, namely Weighted Instance Based Learner (WIBL), to build and handle user profiles. The results of this study were published in IEEE Tenth Jubilee International Symposium on Applied Machine Intelligence and Informatics (SAMI'12).

- This work proposed a novel clustering algorithm and three feature weighting methods for the user profiling. The results of this study have been submitted for publication in a journal.

- This work shows how the Weighted Instance Based Learner (WIBL) algorithm can be used for the user profiling for the provisioning of personalized mobile services. This study was published in Fifth International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services (CENTRIC'12).

- This work investigated well known clustering algorithms and compared their clustering accuracy performance with WIBL in user profiling. The results of this study have been submitted for publication in a conference.

1.3. Outline of the Thesis

The outline of the thesis is as follows.

Chapter 2: User Profiling Methods

The fundamentals of the user profiling and an overview of the user profile methods are presented. The significance of the user profiling for a number of technological methods and applications are discussed. Various user profiling methods, the collaborative, content-based and the hybrid are described, addressing the main techniques and the characteristics. Some of the research works and standards that have been published for user profiling are given. A general discussion on the utilization of the user profiling methods is also given. Two of the well known applications are described as examples of user profiling methods.

Chapter 3: Classification and Clustering Algorithms

Presents classification and clustering for user profiling and evaluates the classification accuracy performance of these classifiers on user profile data. The classification and clustering algorithms that are studied in this chapter are Decision Trees (DTs), Nearest Neighbour (NN) Classifiers, Support Vector Machine (SVM), Bayesian and Naïve Bayesian Networks, Hierarchical clustering, Partitional clustering and Density-based clustering.

Chapter 4: Existing Weighting Methods

In this chapter the feature weighting methods Filter and Wrapper methods are presented. The techniques used for each method have been presented and discussed.

Chapter 5: Proposed Multi-Dimensional Clustering (MDC)

This chapter presents the details of the proposed clustering algorithm and feature weighting methods for user profiling. These are:

1. MDC weight method by Cross Clustering (MDC-CC)
2. MDC weight method by Balanced Clustering (MDC-BC)
3. MDC weight method by changing the Lower-limit to Zero (MDC-LZ)

The simulation results for the proposed algorithm with different user profile datasets are obtained and compared against to the existing algorithms to validate the performances. A case study that implements MDC for a real life scenario is also presented.

Chapter 6: Evaluation, Conclusions and Future Works

This chapter presents a review and evaluation of this thesis, and conclusions are drawn from this research work. Finally suggestions for the future works related to user profiling are given.

Chapter 2

User Profiling Methods

The main challenge in user profiling is the generation of an initial user profile for a new user and the continuous update of the profile information to adapt their changing preferences, interests and needs. The static and dynamic nature of the user related information makes it difficult to retain applicable data within the user profile. In literature two fundamental user profiling methods have been proposed to build and handle user profiles. These are the content-based and the collaborative methods.

In this chapter, overviews on existing user profiling methods are given. Definitions of the fundamental concepts followed by detailed information of the user profiling methods are presented. The disadvantages and advantages of each method are compared and summarized. Two of the well known applications are described as examples of user profiling methods. Finally, related works applicable to user profiling methods, discussions and existing standards are also presented.

2.1. Basic Definitions

In this section the basic definitions of user profiling and relevant terms are described.

2.1.1. User Profiling

The user is an individual or an organization that uses product (i.e. computers) or the services (i.e. web services). The main objective of the product and service providers is to have optimum user satisfaction regarding the quality of service. Technological advances and an increase in the number of products and services lead to user centred developments, which focus on what user want, as well as when and how [6]. Each user is represented with a user profile that is constructed via user profiling. Simply, the user profile is the outcome of the user profiling process (see Figure 2-1).

Figure 2-1 User profile and user profiling

A user profile is a set of information representing a user via user related rules, settings, needs, interests, behaviours and preferences [1]-[4][6]. Hence, a user profile is a collection of personal information. The user information may either be represented as static data (e.g. native country) that is less likely to change or dynamic data (e.g. needs), which is more likely to change overtime.

The content and amount of the information within a user profile can vary depending on the application area. According to Martin-Bautista *et al.* [3] there

are two types of user profiles, simple profiles and extended profiles. Simple profiles include terms extracted from documents that are relevant for the user, while extended profiles, in addition, may contain information about user's educational level, age group, language, knowledge, and country.

Regardless of the information within the user profile, the accuracy of the user profile is based on how the user information is gathered and organized, and how accurate this information reflects the user. Here, the concept of user profiling is needed in order to undertake these activities between the user and the user profile for the maintenance of accurate user profile. According to Oxford Dictionaries Online [5] the definition of the profiling is

> "The recording and analysis of a person's psychological and behavioural characteristics, so as to assess or predict their capabilities in a certain sphere or to assist in identifying categories of people"

whereas user profiling is the process in which the information is gathered, organized and interpreted to create summarization and description of the user [6]. There are two fundamental ways of retrieving information about the user. These are called directly/explicitly or indirectly/implicitly information gathering. In the explicit method, information regarding to the user's interest and preferences, is provided directly/explicitly from the user to the system. For instance, if a web application uses the explicit method to retrieve personal user information then, when each user enters a web site, they may be asked to fill out an online form [12]. Generally, these forms (e.g. online registrations, survey forms or questionnaires) include questions that are aimed to learn the user requirements.

The resulting user profile of the explicit method is referred to as explicit or static user profile. The downside of this method is that explicit profiles have a static nature and are valid only until the user changes their interest and preferences parameters [13]. In the literature, explicit information gathering methods are used by the static profiling that analyzes the static and predictable characteristics of the user.

In contrast, implicit information is gathered dynamically by monitoring the user's interactions with the system automatically. The implicitly created user profile is called implicit or dynamic user profile. Intelligence agents and web-crawlers are examples of the software agents that are used to track the user's behaviour within a website to extract interest and preferences [12]. Also, dynamic profiling uses the implicit method and analyzes user's behaviour pattern (e.g. activities/actions, usage history) to determine user's interests [12] [14]. Hence, the profile data can be updated whenever a user starts a new session (i.e. sign-in to the website). The accuracy of the user profile therefore depends on the amount of generated data. Consequently, the user has to navigate and explore the web site in order for the system to be able to have an accurate profile [12].

It is possible to combine the two methods above and produce a hybrid user profile [12]. The hybrid profile can be achieved in two ways. The first way starts by using the explicit techniques to collect the initial data, followed by the implicit techniques to update the user profile. The second way is in reverse and the implicit techniques first followed by the explicit techniques. In general, it has been cited that the hybrid methods are more efficient than both of the fundamental methods [12]. Table 2-1 [12] summarizes the advantages and disadvantages of all three methods described above.

2.1.1.1. User Profiling Applications

User profiling has attracted a large number of technological methods and applications. Without user profiling, users are treated exactly the same by a system, and it is the first step to find out about the user's needs and expectations. Hence user profiling enables the information professions [6];

- to understand the needs of its users
- to decide what mechanisms and information will be used in order to provide the optimum service delivery, and
- to be aware of the existing constraints

Hence, from an information point of view, user profiling provides a clear understanding on the user's expectations regarding to content, service delivery, filtering, personalizing and customizing information which maximizes the relevance of information provided to the users [6]. In development, marketing and support of the software and games for the mobile phones and devices (i.e. Personal Digital Assistant (PDA), smartphone), user profiling endeavours to provide good quality of service to the customers [15]. To avoid any expensive design mistakes during the product design phase, user profiles can be used to ensure that the design will work for the targeted customers.

Another application of user profiling is within the world-wide-web. It is well known that user profiles can enhance the effectiveness of web mining systems [16]. As described by Martin-Bautista *et al.* [3], user profiling is a key to effective information filtering for web applications where user profile defines customers to online businesses.

One of the main challenges in user profiling applications is the profile initialization for the new user and the continuous updating of the existing profile information based on the user's changing behaviour, interests and preferences. In literature there are two main user profiling methods: content-based [1] [12] and the collaborative [1] [12]. It is also possible to use a hybrid of the two methods [1] [6] [14] which has been detailed in the following sections.

2.1.2. Terminology

In this section the terminology used throughout this thesis will be presented. Personalization, classification and clustering terms as well as the meaning of the terms test instance and training instance are given.

2.1.2.1. Personalization

According to Blom [17] personalization is a process to change the functionality, information content or distinctiveness of a system to increase its personal relevance to an individual. Moreover, personalization is defined as the adaptation of the services in a way that they fit the user's interests, preferences and needs of the user's profile [17]-[23]. From Figure 2-2 it can be observed that the user profile is the input of the personalization process, where services are tailored based on the user profile to meet user's needs and expectations. Hence, the output of the personalization is the personalized service. Generally, there are two types of personalization methods: implicit personalization and explicit personalization. In implicit personalization, information about the user for user profiles is gathered implicitly (e.g. click streams, scrolling, printing and saving) [24]. Therefore, the user is unaware of the information gathering process.

Table 2-1 Comparison of user profile types [12]

	Description	Techniques Used	Advantages	Disadvantages
Explicit User Profiles	User manually creates user profile by means of a questionnaire	Questionnaires Ratings	Preference information gathered is usually of high quality	Requires a lot of effort from user to update
Implicit User Profiles	System generates user profile from usage history of interactions between user and content	Machine learning algorithms	Minimal user effort required Easily updatable by automatic methods	Initially requires a large amount of interaction between user and content before an accurate profile is created
Hybrid User Profiles	Combination of user profile techniques used to create a profile	Explicit/implicit user profiles	To reduce weak points and promote strong points of each of the techniques used	N/A

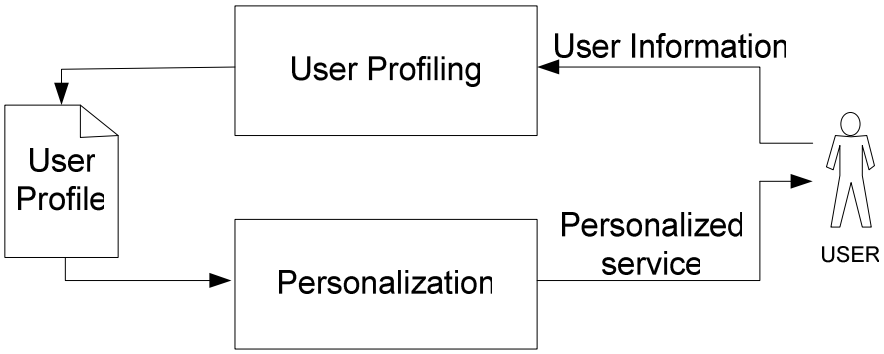


Figure 2-2 User profile, user profiling and personalization

In explicit personalization, on the other hand, user profile information for personalization processes is gathered via direct involvement with the user (e.g. questionnaires, ratings and feedback forms) [24]. Therefore, the user is aware of the information gathering process. In implicit personalization accuracy improves with the continuous use of the system by the user. In explicit personalization, accuracy of personalized information is based on manually provided information that is updated by the user.

2.1.2.2. Classification and Clustering

The term classification is used as an alternative word for the clustering. Nevertheless, there are differences in the meaning of these terms, and therefore they should not be used as interchangeable synonyms.

Classification can be defined as an action of assigning a data object to a class according to the known characteristics of the data object [25]. Clustering, on the other hand, is the process of grouping data objects into the clusters without the prior knowledge of the data objects [26]. Therefore, classification is considered as a supervised learning while clustering belongs to the category of

unsupervised learning. A data object is a set of attributes while classes and clusters are the collection of the instances.

According to Rivero *et al.* [27], classification model, also known as classifier, is a set of patterns which studies the existing data and maps the new coming data one or more classes. Thus, classifiers are using a set of pre-defined or labelled instances to learn a model which can be used to classify the unlabeled instances into one of the pre-determined classes [25]. In the clustering model, conversely, there is no priory knowledge about the clusters and no instances to show the possible relations among the instances [28]. Within same cluster instances are similar between themselves and dissimilar to the instances of other clusters. The clustering model (or clusterer) is described as a set of patterns that studies the existing data and portions it into groups/clusters [27];

Based on the above given information, the differences between classification and clustering are summarised in the following table (see Table 2-2).

Table 2-2 Classification vs. Clustering

Classification	Clustering
Supervised Learning	Unsupervised Learning
Priory knowledge about instances	No priory knowledge about instances
Predefined classes	No predefined classes

2.1.2.3. Symbols

In classification and clustering, instances can be grouped into two: Training instances and Test instances. Training instances is the set of initial information that is used to train the clusterer, while test instances is new information to be clustered.

For example, assume a test dataset with M test instances and a training dataset with N training instances. The test instance vector that corresponds to the i th user and the training instance vector that corresponds to the j th user can be represented as;

$$X_i = \{x_i(1), x_i(2), ..., x_i(A)\}, \text{ for } i = 1,2,3,..., M \quad (2\text{-}1)$$

$$Y_j = \{y_j(1), y_j(2), ..., y_j(A)\}, \text{ for } j = 1,2,3,..., N \quad (2\text{-}2)$$

where, $x_i(k)$ is the value for the k th feature of the i th test instance and similarly $y_j(k)$ represents the value for the k th feature of the j th training instance. Respectively, A is the number of features while the vector of features is $\boldsymbol{f} = \{f_1, f_2, ..., f_A\}$. Here f_k, for $k = 1,2,..., A$, stands for an individual feature which has v_k possible values.

$$f_k = \{f_k(1), f_k(2), ..., f_k(v_k)\} \quad (2\text{-}3)$$

Therefore, $f_k(v_k)$ is the v_k th feature value of the k th feature.

If Q is the number of clusters then the set of clusters is;

$$\boldsymbol{C} = \{C_1, C_2, ..., C_Q\} \quad (2\text{-}4)$$

where C_m is the mth cluster. By the end of the clustering process each test instance is expected to be assigned to a cluster, i.e. $X_j \in C_m$, where m can be any integer from 1 to Q. Here, Q is found by the end of training process.

2.2. User Profiling Methods

This section provides a literature review of the user profiling methods: Collaborative and the content-based (see Figure 2-3).

2.2.1. Collaborative Methods

Throughout the everyday life people seek advice from different resources (e.g. friend and newspaper) to be able to make decisions [12]. A common example can be a friend's suggestion for a summer holiday destination.

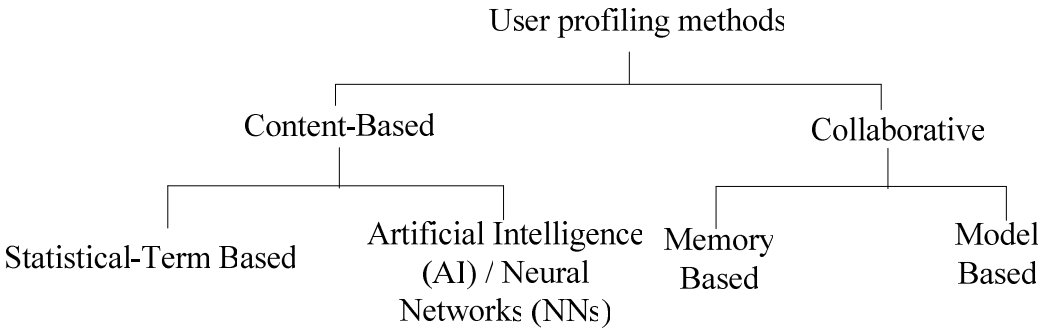


Figure 2-3 User profiling methods

These recommendations affect the way of thinking and help the decision process to be made easier.

The collaborative method has been built on this concept. For user profiling, the collaborative method assumes that the users who belong to the same group

(e.g. of same age, sex or social class) behave similarly, and therefore have similar profiles [1]. The collaborative methods are based on the rating patterns of similar users [2]. In this method people with similar rating patterns, or in other words people with similar taste, are referred to as like minded people [2]. Collaborative methods use filters to build and handle the user profiles. Therefore, these methods are also called collaborative filtering (CF) methods. Here, the term 'filter' corresponds to a criterion that is set depending on the application and the filtering process and decides which information is to be passed on according to the filter in use.

There are two main drawbacks of collaborative filtering: the sparsity and the first-rater problem [12]. The sparsity is the situation when there is a lack of ratings available that is caused by an insufficient number of user or very few ratings per user [12]. The first-rater problem, on the other hand, can be observed when a new user has a deficient number of ratings [12].

Therefore, for example, if a collaborative filtering based recommender system happens to have any one of these issues, then the system can either provide bad recommendations or cannot make a prediction for a user at all. There have been many applications that use collaborative filtering for recommendation purposes. Three of the more popular real applications are the Ringo, the Bellcore and the Grouplens project that was also used as a base for the Movielense recommender [12]. Ringo [12] [29] was published as Firefly and it recommended its subscribers movies and music by making use of collaborative filtering. Similarly, the Bellcore [12] also recommended video films to users by considering their renting patterns.

2.2.1.1. Memory-Based and Model-Based Techniques

Memory-based and model-based techniques enable users to filter the received information according to the ratings, which is the feedback given by the like minded users of the system [30]. Therefore, in these techniques the user can be provided recommendations from the categories which are not previously declared as interesting or relevant by the user but have received high ratings from the users with similar tastes. A user's profile is a set of ratings that the users have given to a selection of items from the system database [2] [30]. As a result, the system's recommendation accuracy improves as the number of ratings increase in a user profile [30].

Figure 2-4 [31] shows the basic principle of the collaborative method. Here, a ratings table is a user-item matrix where each row represents user profile (i.e. j_a) and each column corresponds to an item (i.e. t_h) from the system database.

Systems based on memory-based estimate an item's rating prediction for a particular user (active user/current user), based on the entire collection of previously given ratings by similar users [32]-[34]. There are number of algorithms applied to memory-based systems. The Mean Square Difference (MSD) is one of the popular algorithms where the MSD between the current user profile and all other profiles are calculated. If any user j of the system has MSD below the threshold then that user is considered to have similar taste with the current user. The weight of each user shows the similarity with the current user and calculated as follows [30];

$$w_j = \frac{L - MSD_j}{L} \tag{2-5}$$

where w_j is the weight of the user j and L is the threshold.

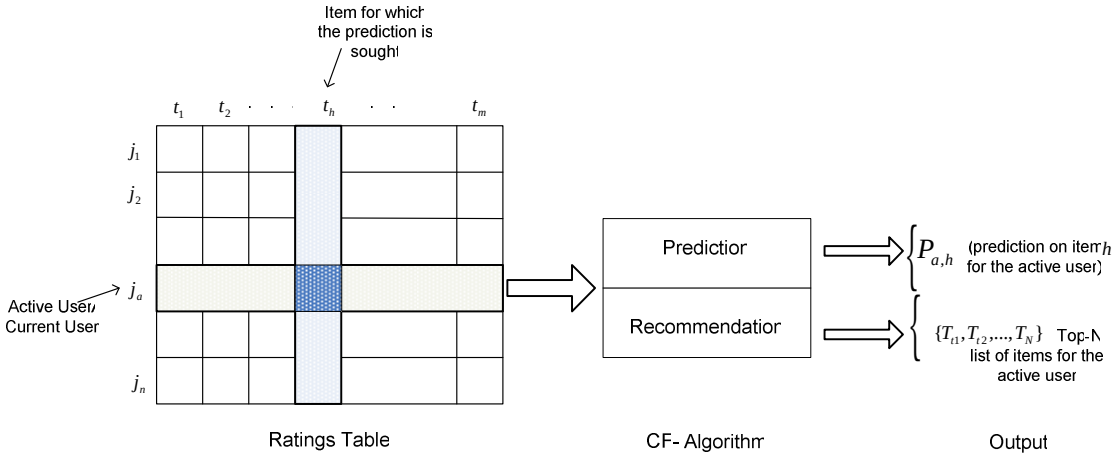


Figure 2-4 Basic principle of the collaborative method [31]

Another popular algorithm to find the user similarity is the Pearson Correlation Coefficient (PCC). The PCC computes the similarity between user j (current user) and i, $w_{j,i}$, as follows [30] [32]-[34];

$$w_{j,i} = \frac{\sum_m (r_{j,m} - \bar{r}_j)(r_{i,m} - \bar{r}_i)}{\sqrt{\sum_m (r_{j,m} - \bar{r}_j)^2 \sum_m (r_{i,m} - \bar{r}_i)^2}} \tag{2-6}$$

where m is set of items that are co-rated by both user j and i while $\bar{r}_j$ is the mean rating of the co-rated items of the current user j. Moreover, $r_{i,m}$ and $r_{j,m}$ show the rating given by the user i and user j to item m respectively. In this measurement if two users give an item the same rating then these users can be identified as similar [32]. The $w_{j,i}$ can have a value between -1 and $+1$.

The positive value indicates positive correlation and shows greater similarity of two users while a negative value is the vice versa [30]. Here, a current user *j* 's rating for a particular item is predicted by taking the weighted average of the known ratings of the similar users [30] [32].

Model-based systems, on the other hand, use the collection of ratings to learn a model that will be used to estimate item rating predictions [32] [33]. Clustering and classification algorithms, which are the topics of chapter 3, are commonly used to make item rating predictions in model-based systems [33][34]. These algorithms treat CF as a classification or clustering problem.

2.2.2. Content-based Method

In an example where a researcher, who works on computer languages, is most likely to search and read articles, books and papers with respect to their subject. Therefore, it is also probable that all these resources have a very similar content. Content-based method is suited to such environments where a user needs items that will match user's preferred content features [12]. Hence, this method has been built on the concept of similarity of contents and assumes that the users show the same particular behaviour under the same circumstances [1]. This method is also referred as content-based filtering due to the use of filters to build and handle user profiles. In this scheme user profiles are represented similar with queries and the system selects the items that have high content correlation to the user profile.

The content dependence is the main drawback of the content-based filtering. Hence, this method performs badly if the item's content is very limited and

cannot be analysed easily by the content-based filtering [12]. Furthermore, eclectic tastes and ad hoc choices also cause bad performance as recommendations made are based on the user's previous choices [12]. For example, consider that a teenage boy who usually buys computer magazines for himself, happens to buy once a travel magazine for his father. In this case, the system may start recommending travel magazines whenever he logs-in.

The following paragraphs describe four different techniques of content-based filtering: Vector Space Model, Latent Semantic Indexing, Learning Information Agents, and Neural Networks Agents.

2.2.2.1. Vector-Space Model

Vector-Space Model (VSM) is a statistical-term based technique and mostly used for the information retrieval. In this model, the contents of various documents are represented with vector/s of weighted terms and the user profile is represented as vector/s of weighted keywords/queries which reflects user's interests and preferences [2]. The dimensions of these vectors are equal to the number of terms that are used to identify the content of the documents or the number of queries that are used to identify the user's interests and preferences [30]. User interests are represented either with a single vector that includes all the interest or with multiple vectors, which reflects interest in several domains [35]. In this model the effectiveness of the user profiles depends on the vector's degree of generalization. The VSM holds both synonym and polysemy issues which may cause unsuccessful detection of the relevant documents and incorrect selection of irrelevant documents. This model assumes that all terms

and related concepts are orthogonal while in reality they are not as a result of synonym [30]. In addition, VSM can only filter text documents.

There are several methods to derive a weighted term representation of the documents or queries. Three of the main methods are Boolean, Term-Frequency (TF) and Term-Frequency Inverse Document Frequency (TF-IDF). The TF-IDF is the most common method. In this method, weight of the term is derived from the number of times that term appears in the document (TF) and inverse of the number of documents in the system that the term appears at least once (IDF) [2]. Consequently, IDF provide high values to the key terms and low values for the common terms. The weight of the term is the product of TF and IDF [30]. Therefore, the weight of term E in S th document D_S , W_{SE} , is given by [36]

$$W_{SE} = TF_{SE} * \log(L / R_E) \quad \text{for} \quad \begin{matrix} S = 1,2,3,...,L \\ E = 1,2,3,...,e \end{matrix} \tag{2-7}$$

where TF_{SE} is the frequency of the term E in S th document D_S . The inverse document frequency of the term E in document collection DC is defined in terms of L and R_E as $\log(L / R_E)$. Here L is the number of documents and R_E represents the number of documents in DC that contains E . The normalization of the weights are calculated as follows [33][36];

$$W_{SE} = \frac{TF_{SE} * \log(L / R_E)}{\sqrt{\sum_{E=1}^{e} (TF_{SE})^2 * [\log(L / R_E)]^2}} \quad \text{so} \quad 0 < W_{SE} < 1 \tag{2-8}$$

The term weights obtained from the equation (2-8), are merged to create weighted term vectors. The similarity between two weighted term vectors is

found by using the well known cosine product, also called normalised inner product [33][34][36].

$$W_{i,j} = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \bullet \vec{j}}{\left\|\vec{i}\right\| * \left\|\vec{j}\right\|} \quad (2\text{-}9)$$

where "$\bullet$" indicates the dot product of two term vectors $\vec{i}$ and $\vec{j}$.

2.2.2.2. Latent Semantic Indexing

Latent Semantic Indexing (LSI) is a statistical-term based technique. This method resolves the orthogonal problem of the VSM by examining the 'latent' structure of a document and the terms within. Singular Value Decomposition (SVD) is one of the techniques that is used in LSI to identify patterns in the relationship between the terms and concepts within a document [30]. Unlike VSM, with the use of SVD, LSI retrieves relevant documents even though they do not have common terms with the user profile [30]. In this technique, the document is taken as a 'word by document' matrix that is computed from the individual document vectors in the system which is obtained using the TF-IDF. This is followed by the reduction of the matrix by typically between 100-300 orthogonal dimensions [30].

2.2.2.3. Learning Information Agents

Learning Information Agents (LIA) is one of the techniques that are used to incorporate Artificial Intelligence (AI) and Neural Networks (NNs) into the user profiling. In this technique, agents use the feedback of the user to update the user profile [30]. Agent technology provides an automated information gathering technique over the internet or any large information repositories (e.g. digital libraries) [3]. Application of the agent technology can be passive filtering of

incoming messages (e.g. e-mail) or active information seeking (e.g. web site detection, browsing assistant, digital libraries search). In LIA the normalised TF-IDF weighting is used to create the vector based representation of the document. In the user profile vector the weight of each keyword corresponds to the user preferences. The learning algorithm that is used by the information agent system uses the selection of documents and associated user evaluation (feedback) to update the weights of the user preferences. If we assume weighted document vector V_i, weighted user profile vector M and user evaluation to page i as e_i then new user profile vector M^+ is calculated as follows [30];

$$M^+ = M + \sum_{i=1}^{p} (e_i \times V_i) \tag{2-10}$$

where p is the number of pages evaluated.

2.2.2.4. Neural Network Agents

Neural Network Agents (NNA) are used to incorporate the AI and NNs into the user profiling and like LIA, user profile updates are made based on the user's feedback. In this technique, user profile reflects the neural network that includes the concepts/terms that are important for the user. The terms in the network are the ones that occur within the documents that are accepted and rejected by the user. In NNA the terms are extracted by using the TF-IDF and they are used to create more comprehensive user profiles. Here, unlike LIA, the user does not have to score the document as the scoring is calculated by the system when a user accepts or rejects the document. Terms are related in the network if the same words are related through the documents [30].

2.2.3. Hybrid Methods

A hybrid method uses both content-based and collaborative methods to counteract the drawbacks [12] [29]. This method guaranties the immediate availability of a profile for each user. The system that employs the hybrid method provides a more accurate description of the user interests and preferences, as it continuously monitors and retrieves the user related information through the system-user interaction [1]. Generally, the hybrid method assigns the new user a default profile with the use of the collaborative method and further enhances the profile using the content-based method [1].

Four hybrid user profiling methods have been introduced in the literature [14]. These are called ‘Static Content profiling’, ‘Dynamic Content Profiling’, ‘Static Collaborative Profiling’, and ‘Dynamic Collaborative Profiling’. The static content profiling is the combination of static profiling and content based methods. Here, the information about user’s interests is gathered during registration. Consequently, in dynamic content profiling, information about user’s interests are retrieved via monitoring user’s behaviour. Moreover, in static collaborative profiling, information relating to user’s interests is collected based on user’s explicit requests. In this method grouping of the users is done explicitly. In dynamic collaborative profiling, on the other hand, information gathering and grouping of users with similar behaviours is done based on dynamic feedback from the users.

Each of the main user profiling methods described above has different characteristics for user profiling. Table 2-3 summarises the main characteristics of these methods [12].

2.2.4. Related works

This section provides an overview of research works and applications that are described in the literature for user profiling. Starting with user profiling for personalized handheld services, personalized web services, personalized television services and real world applications are presented in this section.

In the last decade, personalized services through handheld devices become very popular [7]-[11] [37]. Among those services, many systems have been developed to be used from handheld devices in tourist activities [7][8]. The moreTourism, which stands for "mobile recommendations for tourism" [7], is one of these systems and provides personalized tourist information (i.e. tourist attraction) for users with similar interests. This hybrid system makes use of mashups[1] along with social networks to enhance its users' travelling experiences. To perform recommendation, the social content-based filtering compares the user tag cloud[2] with the attraction tag cloud and the social collaborative filtering creates one new tag cloud for each attraction using the tag clouds of the users who liked it. Hence, the recommendations are based on the user tag cloud, relationship among tags, location in time and space, and the nearby context. According to Lopez *et al.*, the system has been tested with undergraduate students and the preliminary results showed a good performance. Similarly in [8], Fernandez *et al.* proposed a tourism recommender system that offers tourist packages (i.e. include tourist attractions and activities) that best matches the user's social network profiles. Different from [7], the proposed hybrid system does recommendations based on both the user's

[1] A mashup is a hybrid web application that combines sources of information into a new web application [7].
[2] Tags are defined as the collection of keywords which are attached to the web content to describe the content whereas a tag cloud is the collection of tags attached by the users [28].

Table 2-3 User profiling methods

	Description	Techniques Used	Advantages	Disadvantages
Content-based Filtering	Filtering content from a data stream basec on extracting content features that have been expressed in	Vector space model Latent semantic indexing Learning Information Agents Neural Network Agents	Objective analysis of large anc/or complicated (e g multimedia) sources of digital material without much user involvement	1 Content dependent 2 Hard to introduce serendipitous recommendations as approach suffers from "tunnel vision" effect
Collaborative Filtering	Filtering items based on similarities between target user's collaborative profile and peer user/group	Memory-based Mode -based	1 Content independent 2 Proves more accurate thar content-based filtering for most domains of use enables introduction of serendipitous choices	1 Sparsity poor prediction capabilities when new item is introduced to database due to lack of ratings 2 new user poor recommendations made to new users until they have enough ratings in their profiles for accurate comparison to other users
Hybrid Filtering	Combines two filtering techniques	Collaborative Content based	To reduce weak points and promote strong points of each of the techniques usec	Weak points can out-weight strong points if the hybrid is created naively

viewing histories (Digital Television (DTV) viewing histories received from the user's set-top boxes via a 2.5/3G communication network) and the preferences in the social network (i.e. preferences of the user's friends). The system has been tested on 95 users and according to the evaluations, 81% of the users appreciated the recommended tourist attractions and contributed to spread the offers to their friends through the social network while 90% are willing to pay for such a personalized recommender system on social network.

Since the amount of resources and information on the web is vast, personalized web services and user profiling has become more important for web users. Various works has been carried out to address online service personalization [38]-[43]. In [39], Yeung *et al.* proposed a technique to analyse the personal data, personomies, within the folksonomies[3]. This work aimed to investigate how accurate the user profiles can be generated from the folksonomies and discuss how these profiles can be used for the web page recommendation. The proposed algorithm aimed to generate user profiles that were representing user's multiple interests. The method was tested on the data which was taken from the del.icio.us[4] web site. This data was the collection of bookmarks and tags that have been used by the users. Here, the vector space model has been used for the term vector representation of tags, bookmarks (documents) and queries. The cosine similarity has been used to find the similarity between the bookmarks and the queries and the evaluation is done based on precision, recall and F1[5] measures. In [38], Park *et al.* proposed a hybrid framework for online video recommendations where the recommendations are done according

[3] Folksonomies are the user-contributed data that are collected via collaborative tagging systems [39].
[4] www.delicious.com
[5] F1 is the harmonic mean of precision and recall.

to the similar viewing pattern. In this work, user profiles are constructed as an aggregate of tag clouds, also known as global tag cloud[6], of videos. Here, user profiles and videos were represented with tag cloud vectors. The cloud-based cosine similarity was employed to compute the user similarity. Here, the user's profile is updated every time the user plays a video, by including the global tag cloud of the video into the user's tag cloud. Park *et al.* argued that different from the existing hybrid methods, this approach is based on the implicit users' view-transaction data instead of the explicit ratings data. Another hybrid framework has been proposed in [40]. Different from the works describes above, in [40] collaborative filtering was employed together with techniques from the Multi Criteria Decision Analysis (MCDA)[7] for item recommendation. In this study user profiles were included with user's numerical ratings and ranking order, and represented as vectors. The user profile is updated with a feedback mechanism, which is activated by the user when he/she is willing to rate an item after a recommendation. In this system the MCDA was used to find the similar users while collaborative filtering was used to recommend items.

There has been a considerable amount of work for personalized program and advertisement recommendations for television (i.e. for Internet Protocol Television (IPTV) and Integrated Digital Television (iDTV)) users [44]-[47]. In [44], a hybrid TV program recommender system, gueveo.tv, has been proposed. According to the Martinez *et al.*, the proposed system works well because both methods are complement with each other in a way that the content-based method recommends usual programs and collaborative method

[6] Here, the global tag cloud of a video is constructed by aggregating all the tags that all the users have attached to the video.
[7] Multiple Criteria Decision Analysis (MCDA) is a well established field of decision science that aims at analysing and modelling decision makers' value systems to support them in the decision-making process [40].

provides the discovery of new shows. In this study, each user represented with user's preference profile that contains two types of information that are domain preferences (i.e. list of available TV channels, preferred viewing times) and program preferences (i.e. subject keywords or tags). This information was gathered via implicit (i.e. monitoring viewing times) and explicit methods (i.e. filling questionnaire). In gueveo.tv, vector space model has been employed to generate a vector representation of the user profile and programmes viewed. Here, cosine measure is used to calculate the similarity between the program vectors and the user profile vectors. The system has been tested with real users and results were shown as positive [44].

Amazon.com and Yahoo! Music are two popular real world applications that use content-based and/or collaborative methods.

Amazon.com employs a content-based method for collaborative filtering. In this hybrid application the content-based method finds the relationships between the items so that the system can recommend items that are similar to other items the user has already bought (see Figure 2-5 [48]). A user's response to these recommendations is then utilized by collaborative filtering to compute recommendations based on like-minded people (see Figure 2-6 [48]).

Yahoo! Music utilizes content-based method for music recommendations. Here, the system tracks a user's watching, listening and rating patterns to model a user's preference profile. In this website a user can browse music by videos, songs, albums and artists.

Figure 2-5 Recommendations based on content-based methods [48]

Figure 2-6 Recommendations based on collaborative method [48]

Each music piece is presented with rating options (see Figure 2-7 [49]). The system uses the provided ratings to find similar content to recommend (i.e. music from similar; artist and music category). Hence, to get the best recommendation users have to rate more music.

Figure 2-7 Music video recommendation on Yahoo! Music [49]

Yahoo! Music recommends music based on just the user's rating pattern. Here, the user profiles is derived from the explicit profile techniques. These techniques can provide higher quality personal information to the system than the implicit techniques. However, they require a lot more effort from the user to update the preference information. Amazon.com is able to provide an item recommendation based on purchase history of a customer and the like minded people. In this system user profiles are conducted using both explicit and implicit profile techniques. With the implicit method a vast amount of data can be gathered at no extra cost to the user (i.e. cost of providing feedback) which makes the implicit method an attractive alternative over explicit method. A more informed knowledge of customer's preferences is obtained using hybrid filtering methods together with hybrid profiles. A personalized experience can be available when more detailed information is known about the customer, however building and maintaining such a system can be very expensive. Although Yahoo! Music is a good example for content-based filtering, using hybrid profiles may increase the flow of the recommendations and decrease the required effort from the user.

2.2.5. Discussions

From the Subsection 2.2.4. it can be seen that collaborative and content-based methods have been widely used for the personalization in various applications. Here, the content-based systems have mostly been designed to recommend text-based items (i.e. documents in www) via predicting ratings or the relative preferences of the user (i.e. ranking order). In these systems, user profiles are mostly described with keywords obtained by analysing the items which have been previously seen or rated by the user. These applications also showed that the user profile can be represented as a vector of weighted keywords, where the cosine similarity is commonly used.

The collaborative systems are mostly used for e-commerce websites and they consider similar buying behaviours of the customer, to estimate a particular user's preference on items. In these systems, the user profiles retain the ratings of items which other users have already rated. This is achieved by the cosine similarity and Pearson correlation (similarity measurement techniques) which identifies the similarity between users. The cosine similarity is utilised both for content-based and collaborative systems. Yet in content-based it is used to find the similarity between the term vectors, while in collaborative systems it is used to find the similarity between the vectors of actual user ratings.

As previously discussed in Section 2.2. and in Table 2-3, both collaborative and content-based methods suffer from many limitations. However, hybrid systems have been proposed to overcome these limitations via utilizing both methods. It has been observed from the current hybrid systems that the content of the user profiles are just maintained.

More recently, tag aggregation based personalization has received considerable attention and in current studies user profiles are represented with tag clouds. It can be argued that this way of representation tackled the previously mentioned sparsity and first-later problems (see Subsection 2.2.1.). This is because, similarities between users does not have to be calculated based on user's common ratings. Moreover, tags make it unnecessary to analyse the content of the web page, video and advertisement, which can be a difficult process to build user profiles. Hence, it can be argued that this offers a solution to a content dependence limitation of the content-based method (see Subsection 2.2.2.). However, in these systems, the quality of the user profiles rely on the number of users participating in tagging and the number of tags the user used that are produced by others. Hence, tag cloud based user profiles reflect the web content more than user itself.

Accurate user profiles are important to both the user and the service provider. From the user point of view it is important for the personalized services not to be misrepresented. For the service providers, on the other hand, it is the way to achieve optimum user satisfaction by providing accurate personalized services.

It can be seen from the above sections (Subsection 2.2.4. and 2.2.5.), the literature on user profiling focused on the usage of profiling features such as ratings, items, keywords and simple demographics to represent each user. Although this traditional way of profiling works well for specific services, it lacks in representing the multidimensionality of the user profiles accurately. For example, user profiles that reflect the ratings which were given to music videos cannot be used to recommend books for the same user. This constraint motivates the need to conduct more advance profiling to build a more

comprehensive profiles to describe user's interest, preferences and demographics. This way of profiling can provide user related information that can be used by various third party service providers for different service personalization.

To be able to use the multidimensional profiles effectively, feature weighting should be taking into account. Utilization of feature weighting is therefore essential for accurate user profiling. This is because the relevancy of all information contained within the user profile is not the same for different service personalization. For example, user's book interest information may not be as relevant as income information of the user for personalized restaurant recommendations. Using weights to make the distinction between relevant and irrelevant information can provide a solution for this problem.

It may be concluded from the above explanations, current user profiling works when it;

- does not consider multidimensional structure of the user profile, and
- does not apply feature weighting for the user profiling.

To address these problems in the following chapters of this thesis different classification and clustering algorithms and feature weighting methods for multidimensional user profiling are investigated.

This research will be the first in the literature to address multidimensional structure of the user profiles and feature weighted user profiling to create accurate user profiles.

2.2.6. Standards and Projects

The emphasis of user profiling for different application areas and methods has led to a search for new standards and projects for user profiling. The Composite Capability/Preference Profiles (CC/PP), Universal Agent Profile (UAProf), Generic User Profile (GUP) and the Liberty Alliance are some related works.

2.2.6.1. Liberty Alliance

The Liberty Alliance project is an alliance of more than 150 companies (i.e. AOL, ORACLE, British Telecommunications plc (BT)), non-profit organizations (i.e. SAFE Bio Pharma) and governments (i.e. U.S. Department of Defence, New Zealand Government State Services Commission). It is aimed to develop an open standard for management of federal network identities that supports all current and emerging network devices. In the project's architecture, Liberty Alliance defines a role called Attribute Provider, and specifies how the access to such an attribute provider should be implemented in a standard manner. It also specifies a protocol that can be used between a Service Provider and Attribute Provider, which allows the sharing of user profile data called attributes (e.g. preferences and settings) [50].

2.2.6.2. Composite Capability/ Preference Profiles (CC/PP)

The CC/PP is a system that is developed by World Wide Web Consortium (W3C) and describes device capabilities (hardware and software) and user preferences. The concept behind this system is to have universal access to the Web with whatever terminal people are using. Universal Agent Profile (UAProf) by Open Mobile Alliance (OMA) implements CC/PP which allows proxies to transform content to mobile devices that are supporting Wireless Application

Protocol (WAP) and JRS 188 (Java Specification Request). CC/PP Processing defines a set of Java Application Programming Interface (API) for processing of CC/PP and UAProf documents [19].

2.2.6.3. ETSI Human Factors: User Profile Management

This project defines the requirements for user profile management. According to the final draft of the project, profiles can be used to improve communications for young people and people with various disabilities, while it should still be sufficient for ordinary people. In this project's draft, the detailed information about the concept of user profile (i.e. profiles and the existing profile types) is given. According to the report, European Telecommunications Standards Institute (ETSI) does not propose a framework or detail specification, but do provide recommendations on personal profile management and what it should consist of [3] [50].

2.2.6.4. Generic User Profile (GUP)

GUP is defined by 3rd Generation Partnership Project (3GPP). According to 3GPP, GUP is the collection of user-related data which define how an individual user access and experiences services. The aim of this concept is to define flexible and extensible user profiles that can be accessed and managed by different stakeholder/s using a standardize access mechanism in the mobile network [22][51]. In GUP, data can be stored in a home network and in value added service provider equipment allowing intra-network and inter-network usage [50]. In intra-network, data is exchanged between applications within a mobile operator's network while in inter-network, this exchange is carried out between the mobile operator's network and the value added service providers

[22]. Therefore, the GUP provides the access of data for ranges of services and functions.

2.3. Summary

In this chapter, the fundamental user profile and user profiling definitions were presented. The terminology for personalization, clustering, classification, training instance and test instance was given. This included the clarification of the user profile and user profiling. The relationship between these concepts, different user profile types as well as the techniques to achieve these profiles was also discussed in this chapter. The need for the user profiling for a number of technological methods and applications have been described.

Moreover, various user profiling methods, the collaborative, content-based and the hybrid, have been described. The characteristics and techniques for each method have been presented. The drawbacks and advantages were discussed and summarised in a table. Some of the research works and standards that have been published for user profiling were described and included in this chapter. General discussions on the utilization of the user profiling methods were also given. Two of the well known applications have been described as examples of user profiling methods.

The use of user profiling for more comprehensive applications led to new studies. For this reason, Machine Learning (ML) and Data Mining (DM) methods have been introduced to the user profiling which will be discussed in the next chapter.

Chapter 3

Classification and Clustering Algorithms

Classification and clustering algorithms are widely used in Machine Learning (ML) and Data Mining (DM) applications. In personalization applications the main aim of the ML methods is to reduce the need for user interaction for the purpose of user profile updating. On the other hand, the purpose of the DM methods in personalization applications is to extract useful information from the vast amount of user related data sets or databases [52].

This chapter investigates the well known classification and clustering algorithms. The classification process and the most popular classification algorithms such as Decision Trees (DTs), Nearest Neighbour (NN) Classifier, Support Vector Machine (SVM) and Bayesian Classifier are explained. Moreover, clustering algorithms (i.e. hierarchical clustering, partitional clustering and density-based clustering) are described in some detail. Finally, the use of the classification algorithms for user profiling and the simulations that were carried out on different classification algorithms will be presented.

3.1. Classification

Classification is a supervised learning and is one of the commonly used DM tasks. As DM became more popular, the use of classification algorithms within different applications has increased [25]. Classification can be thought of as a process that analyses a set of data to build a distribution model, which is then used to classify the newly presented data. Hence, the classification process has two steps [53]. In the first step, a set of data, i.e. training data, is used to build a classification model that matches the training data with user predefined classes [53]. Following this, in the second step, the new test data is classified using the constructed model [53]. Here, training data includes pre-classified instances while test data is a set of un-classified instances. A number of classification algorithms have been proposed in the literature. Figure 3-1 shows the more popular techniques for classification [25] [53] [54].

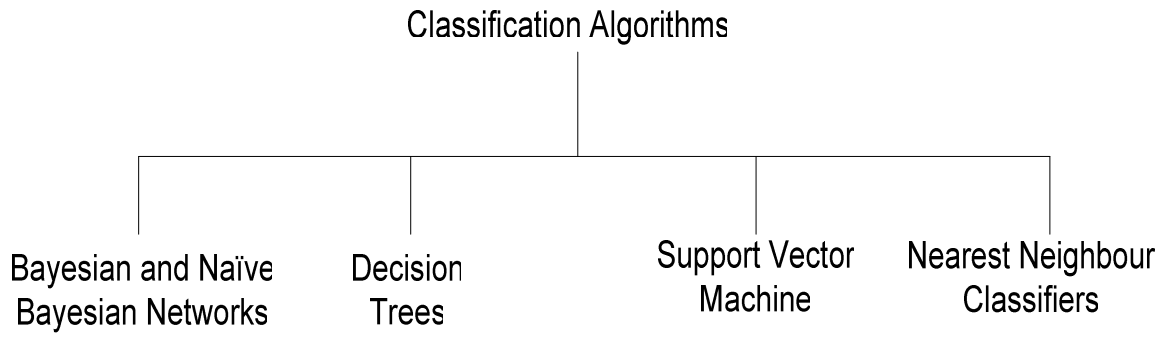


Figure 3-1 Classification algorithms

3.1.1. Bayesian and Naïve Bayesian Networks

Bayesian Networks (BN) is one of the well known classification algorithms that is named after Thomas Bayes (ca. 1702–1761), founder of the Bayesian

methods. BNs are probability values, which are based on and used for the reasoning and the decision making in uncertainty where such reasoning heavily relies on Bayes' rule [55]. Bayes' rule can be defined as follows [55]:

- Assume Cl_b as class label where $Cl = \{Cl_1, Cl_2, ..., Cl_B\}$ and $b = 1,2,..., B$.
- Based on the Subsection 2.1.2.3., assume X_i as unclassified test instance where $X_i = \{x_i(1), x_i(2), ..., x_i(A)\}$ for $k = 1,2,..., A$.
- X_i will be classified into class Cl_b with the maximum posterior class probability $P(Cl_b \mid X_i)$,

$$P(Cl_b \mid X_i) = \arg\max_{Cl_b} P(Cl_b) P(X_i \mid Cl_b) \quad (3\text{-}1)$$

Hence, the basic prerequisites of the BN calculation are [53];

- The knowledge of the prior probability for each class Cl_b
- The knowledge of the conditional probability density function for $P(X_i \mid Cl_b) \in [0,1]$

BN can represent uncertain attribute dependencies. However, BN has high computational complexity, so it is Non-deterministic Polynomial (NP) hard to learn optimal BN [53][56]. Moreover, BN needs complete knowledge of prior and conditional probabilities [53]. Figure 3-2 [55] shows a basic BN representation.

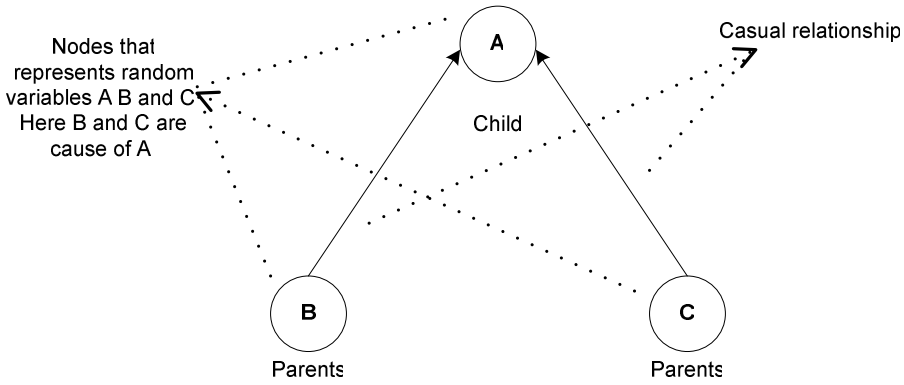


Figure 3-2 Basic bayesian network [55]

Naïve Bayesian (NB) classifier is one of the Bayesian classifier algorithms. In many works it has been proven that NB classifiers are one of the most computationally efficient and simple algorithms for ML and DM applications [57]-[61]. Unlike BN, NB classifiers assume that all attributes within the same class are independent, given the class label (see Figure 3-3 [58]). Based on this assumption, which also reduces the computational complexity of BN classifier, the NB classifier modifies the Bayesian rule as follows [55] [58]:

$$P(Cl_b \mid X_i) = \arg\max_{Cl_b} P(Cl_b) P(x_i(1), x_i(2), ...x_i(A) \mid Cl_b)$$
$$P(Cl_b \mid X_i) = \arg\max_{Cl_b} P(Cl_b) \prod_{k=1}^{A} P(x_i(k) \mid Cl_b) \quad (3\text{-}2)$$

The balance between efficiency and effectiveness, thus the balance between cost and the learning process and the quality of the learned model, with the expressive power, make NB networks a good candidate for interactive applications [59]. Nevertheless, because of its naïve conditional independence assumption, optimal accuracy cannot be achieved. For this reason, a number of algorithms have been developed to increase the accuracy in NB [56] [58] [59].

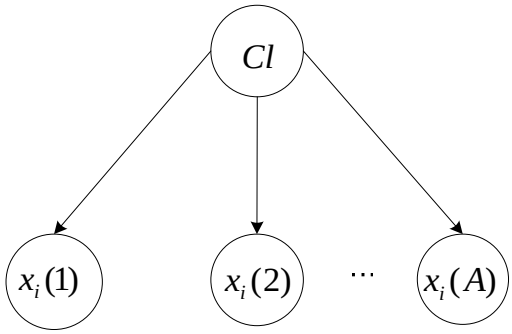


Figure 3-3 Naive bayesian classifier [58]

The Lazy Learning of Bayesian Rules (LBR) is one of the lazy learning algorithms that have been proposed to improve the accuracy performance of the NB. The LBR algorithm applies lazy learning techniques to the NB rule [57]. At the classification time of each test instance, LBR builds the most appropriate Bayesian rule for the test instance.

3.1.2. Decision Trees

Decision Trees (DTs) are data structures that can examine the data and induce the tree and its rules to make predictions [62]. A successful classification with the DTs requires well-defined classes and pre-classified training data [53]. The classification accuracy on the training data set and the size of the tree affect the quality of the DT.

Construction of the tree model incorporates two-phases; building phase and pruning phase. The building phase includes a series of division on training dataset that is carried out based on the decision rules [53]. This partitioning is continued until the resulted classes have homogenous instances. In the pruning

phase, on the other hand, the nodes that may cause over fitting and low accuracy are pruned [53]. Figure 3-4 shows the illustration of decision tree [53].

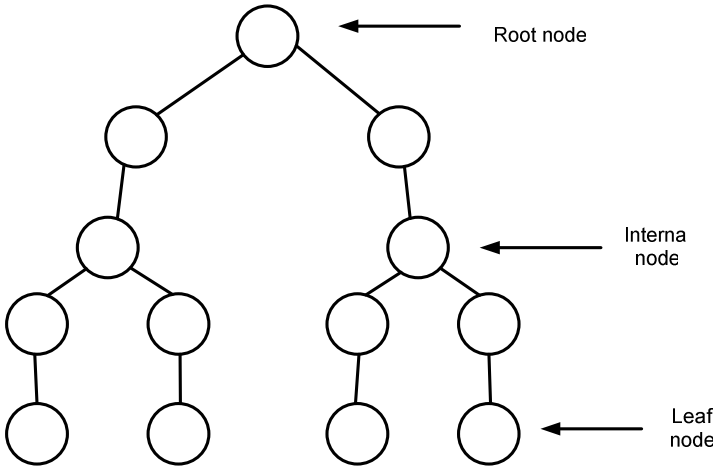


Figure 3-4 Illustration of decision tree

In the above figure, the Root node is the class attribute chosen from the dataset to be used as a base to build the tree upon [53]. The Internal node is an attribute that resides in the inner part of the tree [53]. The Leaf node is one of the predefined classes [53].

After the building phase of the tree model, the DT classifier is ready to classify the test instances. Here, each instance enters the root node to be classified. The root node decides which internal node the instance will be placed next [63]. Although this initial decision can be changed based on the chosen algorithm, the aim is to find the best suited class for the new example [63]. This classification process finalizes when the instance arrives to a leaf node. All the instances within the same leaf node (class) are following the same unique path from the root to leaf node [63]. This path is the expression of the decision rules that have been used for the classification [63]. Following Figure 3-5 [64] is an example of the classification process of the DT.

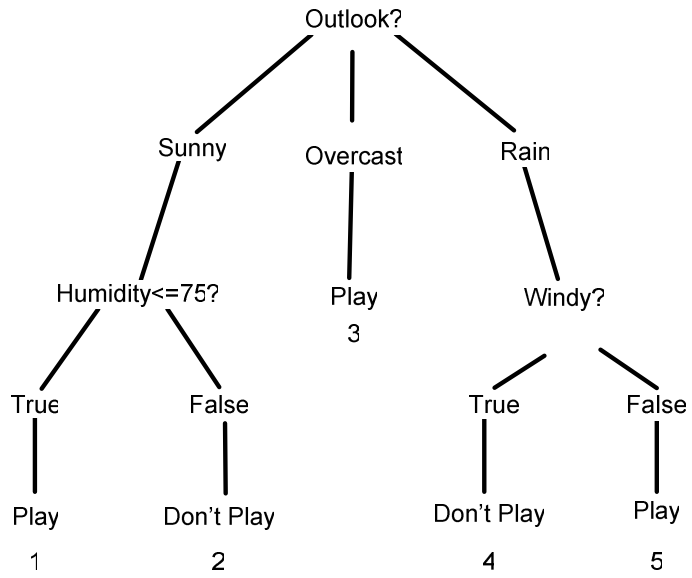


Figure 3-5 Decision tree to classify days as play or don't play [64]

The DT in Figure 3-5 classifies days as play or do not play. In this example it is assumed that a weather of a particular day represented with the following attributes and attribute values [64] in Table 3-1;

Table 3-1 Attribute and attribute values

Attribute	Attribute value
Outlook	Rain overcast sunny
Temperature	Continues values
Humidity	Continues values
Windy	True false

While the test instance to be classified have the attribute values as outlook=sunny, temperature=60^{o}, humidity=70%, windy=true, according to these values the test instance will be classified into the 1^{st} leaf node.

Furthermore, there are a number of DTs in the literature. The Iterative Dichotomister 3 (ID3), C4.5 and Naïve Bayesian Tree (NBTree) are some of the most popular DT algorithms.

3.1.3. Support Vector Machine (SVM)

Support Vector Machines (SVMs) are supervised learning methods that are used for classification. These methods perform classification by constructing an N-dimensional hyperplane that optimally separates the data into two classes [65] [66]. In hyperplane each example is represented as a positive or a negative point (see Figure 3-6 [67]). The aim is to have the maximum separation margin between these positive and negative examples so as to minimise training dataset error (empirical risk) and generalization error (test dataset error or confidence interval) [25][54]. Here, the Support Vectors (SVs), which are a small fraction of the training data, are used to define the dividing line between two classes (see Figure 3-6 [67]).

As a classifier, initially, SVM takes a set of examples as an input and performs a prediction to match each example with one of the two classes. Therefore, this input set is used to train the SVM classifier to build the prediction model that will predict whether the new example, i.e. test data, belongs to a negative or the positive class. Here, the input set, i.e. training dataset, have the labelled examples, where each example is a member of one of the two classes.

The SVM has a high performance in practical applications such as text classification and pattern recognition [25].

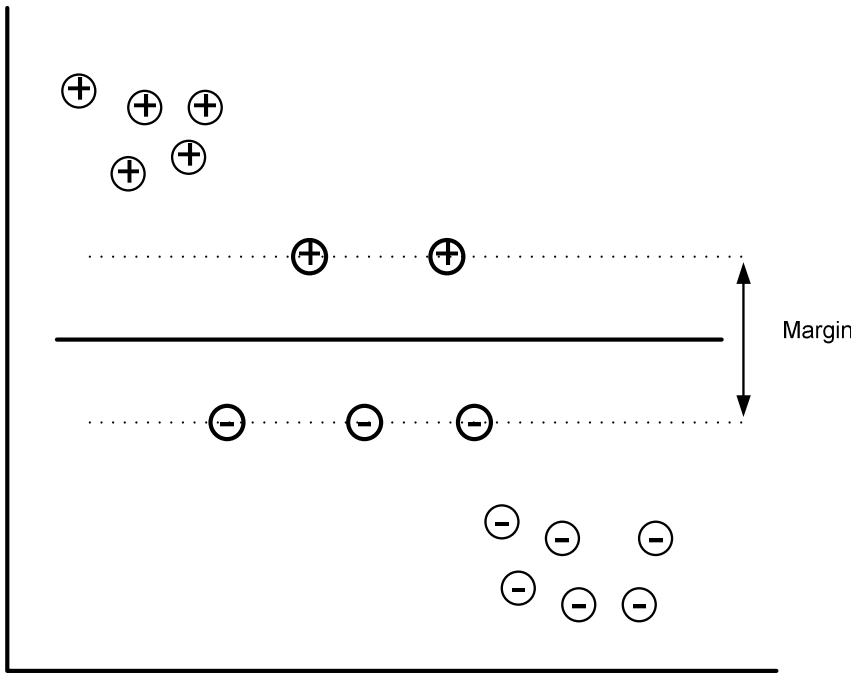


Figure 3-6 Support vector machine model [67]

3.1.4. Nearest Neighbour Classifiers

Nearest Neighbour (NN) algorithms have been widely used for classification problems. In NN classification, each new test instance is compared with the training instances using normalized Euclidean distance and the closest training instance is predicted to have the same class label with the test instance [68]. In case of several training instances qualified as the closest, the first one is used [69]. Instance Based Learner (IBL) is a comprehensive form of the NN algorithm, which normalizes its features ranges, processes instances incrementally and has a simple policy for tolerating missing values [69].

The comparison between the test instance X_i and the training instance Y_j is performed feature by feature where:

If the kth feature is numeric,

$$g(x_i(k), y_j(k)) = (x_i(k) - y_j(k))^2 \tag{3-4}$$

else if the kth feature is symbolic,

$$g(x_i(k), y_j(k)) = \begin{cases} 0, if\ x_i(k) = y_j(k) \\ 1, if\ x_i(k) \neq y_j(k) \end{cases} \tag{3-5}$$

where $g(x_i(k), y_j(k))$ is the function showing the similarity between the kth feature values of the instances X_i and Y_j.

In the IBL algorithm the similarity of the two instances is defined by evaluating the distance between their corresponding feature values, which can be found as:

$$dist(X_{i,}Y_j) = \sqrt{\sum_{k=1}^{A} g(x_i(k), y_j(k))} \tag{3-6}$$

The IBL aims to assign the cluster label of the training instance, which is closest to the test instance of interest in terms of (3-6), i.e. the decision criterion is $\arg\min_j dist(X_i, Y_j)$ for $i = 1,2,3,...,M$.

3.2. Clustering

Clustering, also called unsupervised classification, is the process of segmenting heterogeneous data objects into a number of homogenous clusters [63]. Each cluster is a collection of data objects that are similar to one another and dissimilar to the data objects in other cluster/s [54]. A successful clustering

algorithm has clusters with high intra-class similarity and low inter-class similarity [54] (see Figure 3-7 [53]).

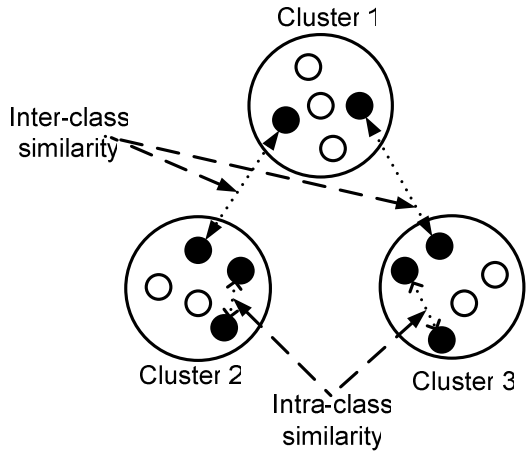


Figure 3-7 Intra and inter cluster similarity [53]

Each clustering algorithm uses a different method to cluster the information. In the literature the most popular clustering methods can be categorised as shown in Figure 3-8 [54] [70].

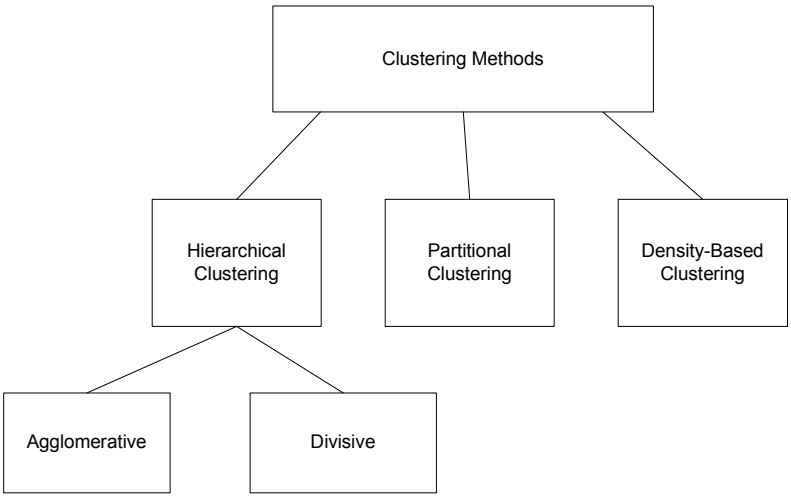


Figure 3-8 Clustering methods

3.2.1. Hierarchical Clustering

Hierarchical clustering is the process to create a hierarchical decomposition (dendogram) of the set of data objects [54]. Hierarchical clustering performs either a merger of clusters (Agglomerative method) or division of a cluster at the previous stage (Divisive method).

In Agglomerative method, initially each data object describes a cluster, and then recursively clusters are merged together until only one cluster remains. In Divisive method, on the other hand, initially all data objects describe one cluster, and then recursively large clusters are divided into smaller clusters. Figure 3-9 shows the dendogram of the six data objects with top-down (divisive) and bottom-up (agglomerative) methods [54][71][72].

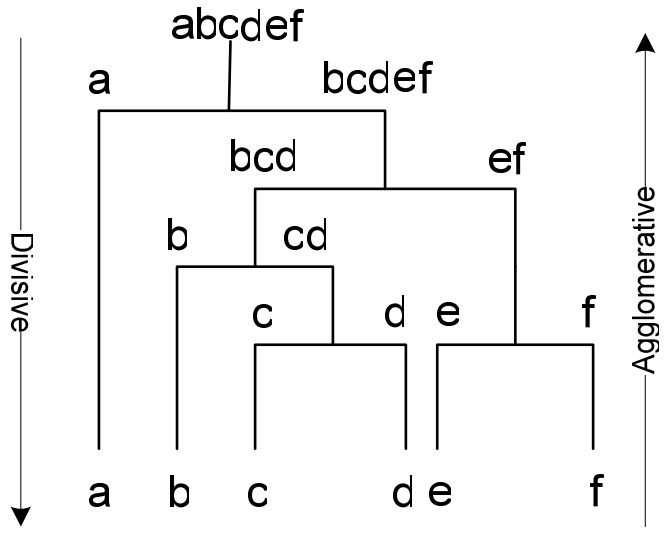


Figure 3-9 Illustration of hierarchical clustering and the agglomerative and divisive methods

The well known hierarchical clustering algorithms are; Single-linkage, Complete-linkage and Average-linkage. Linkage clustering methods have reasonable clustering results with real-world data sets [73].

In single-linkage clustering the resulted distance between one cluster and another cluster is equal to the shortest distance from any member of one cluster to any member of the other cluster [72]. Here, the shortest distance reflects the maximum similarity between any two data objects in two different clusters. The single-linkage algorithm is also called the nearest neighbour, connectedness or minimum distance method [71][72].

The complete-linkage clustering is the opposite form of the single-linkage clustering since in complete-linkage the link between two different clusters is expected to be the maximum distance from any data object of one cluster to any data object of the other cluster [70]. The maximum distance reflects the minimum similarity between two data objects in two different clusters. The complete-linkage algorithm is also called farthest neighbour, diameter or maximum distance method.

The average-linkage clustering can be thought as a combination of single and complete-linkage algorithms. Here the link between two clusters is equal to the average greatest distance of all paired data objects of these clusters.

3.2.2. Partitional Clustering

Partitional clustering is a non-hierarchical clustering method. This method creates disjoint clusters in one step by decomposing the dataset. Therefore, there is no relationship among the clusters [12].

K-means is the most representative algorithm of partitional clustering [53]. In this algorithm the number of clusters, Q, is defined by the user. Then, randomly selected Q data objects become the center (cluster centroid) of the Q clusters. The rest of the data objects are assigned to the closest clusters. Here, the cluster center is represented by the mean values of the data objects within the cluster. Therefore, every time that the cluster centroid is being updated a new data object becomes a member of a cluster. This process is repeated until no change can occur. Following figure is an example to summarize convergence of K-means clustering algorithm as defined above. Here $Q=2$ [74].

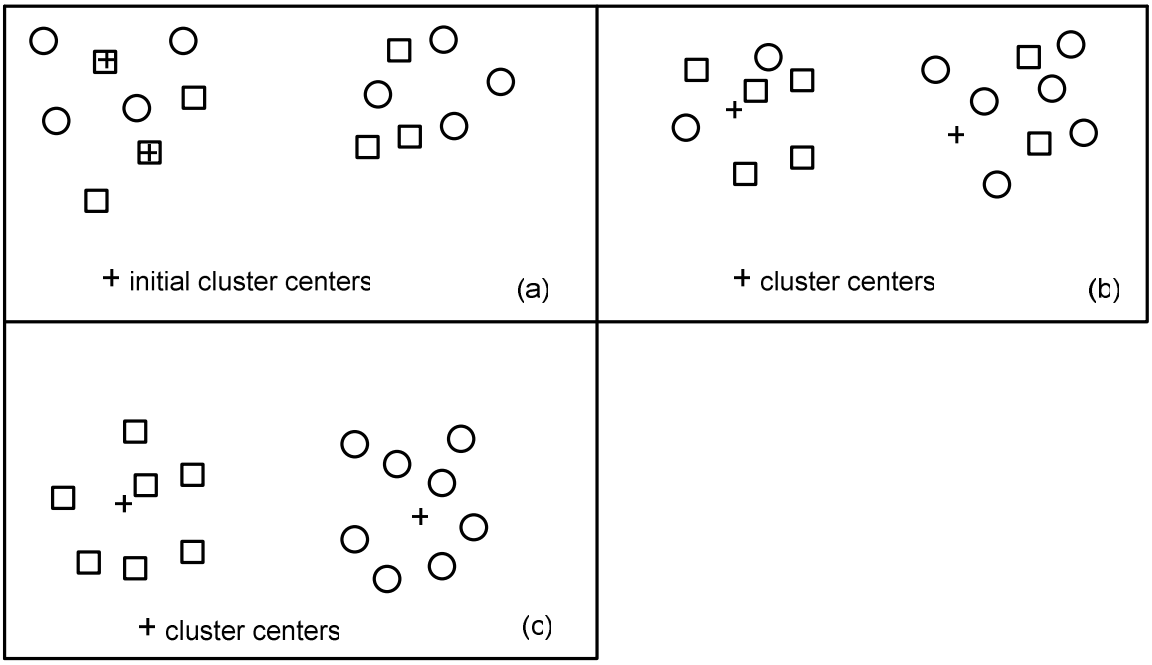


Figure 3-10 Convergence of K-means partitional clustering: (a) first iteration; (b) second iteration; (c) third iteration [74]

3.2.3. Density-Based Clustering

Clusters have various sizes and shapes. Clustering based on the similarity distance between the data objects, results only spherical shaped objects. To

find clusters with complex shapes requires a more comprehensive method than partitional clustering methods. Density-based clustering methods have been developed to find the clusters with arbitrary shapes. Such methods use connectivity and density functions to find arbitrary shape clusters [54]. In the data space, these methods consider clusters as dense regions of data objects which are separated by low density regions [25]. A good example for the density-based clustering is the Density Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm. Such an algorithm can be used to filter the noise and to find arbitrary shape clusters within the datasets. The idea behind the DBSCAN is to grow the given cluster as long as the nearest neighbours exceed some threshold [25]. This means, for each data object within the cluster, there must be at least a minimum number of data object (neighbours) for a given radius [25].

Based on the aforementioned information, Table 3-2 summarises the characteristics of clustering methods [75]. In this table time and space complexity are represented with three parameters where the number of patterns to be clustered is N, the number of clusters is Q and L is the number of iterations [76] [77].

3.3. Classification in User Profiling

Major classification algorithms were explained in detail in the previous section. In this section utilization of classification algorithms to classify user related information to create accurate user profiles is described.

In the literature there seem to be a lack of comparison of these algorithms with classification accuracy of the user profile information. For example, in [78],

Panda *et al.* compared the performance of NB, Id3 and J48 algorithms for network intrusion detection. According to the simulation results NB performed better than Id3 and J48 with respect to overall classification accuracy. However, Panda *et al.* added that, in comparison to NB, DTs (Id3 and J48) were robust in detecting new intrusion/attacks. In [79], Zhang *et al.* compared the ranking performance of NB classifier with the DT (C4.4) classifier. The experiments were conducted using 15 datasets from University of California Irvine (UCI) data repository [80]. According to the experimental results, NB algorithm outperforms the C4.4 [81] algorithm in 8 datasets, ties in 3 datasets and loses in 4 dataset. The average Area Under Curve (AUC) of NB is 90.36% which is substantially higher than the average 85.25% of C4.4. Considering these results, Zhang *et al.* argue that NB performs well in ranking, just as it does in classification.

In another work [82] Huang and Ling compared the accuracy and AUC measures for learning algorithms and claimed, both formally and empirically, that AUC was a better measure than accuracy. They re-evaluated the well known ML algorithms based on accuracy using the AUC measure. The experiments were conducted two times. The first experiment was conducted on three kinds of artificial datasets which were binary balanced, binary imbalanced, and multiclass. The second experiment was conducted on 18 real-word datasets with relatively large number of examples from the UCI data repository. For the second experiment C4.5, C4.4, NB and SVM learning algorithms have been used. According to the experimental results, average predictive AUC values of NB, C4.4 and SVM were found to be very similar. Wang *et al.* [57] compared and constructed the relative performance of LBR and Tree Augmented Naïve Bayesian (TAN). In this work the TAN algorithm was used to

approximate the interactions between attributes by using a tree structure imposed on the NB structure [58]. LBR is desirable when small numbers of data objects are to be classified while TAN is desirable when large numbers of data objects are to be classified [68].

In [56], Jiang and Guo proposed the Lazy Naïve Bayesian (LNB) algorithm and compared it with Selective Neighbourhood based Naïve Bayesian (SNNB), Locally Weighted Naïve Bayesian (LWNB) and LBR. According to the presented work, SNNB and LWNB improved the classification accuracy of NB while LNB improved ranking accuracy of NB by 0.92%. LNB was found to spend no effort during training time and delay all computation until classification has started. LNB learning algorithm deals with NB's unrealistic attribute conditional independence assumption by cloning each training instance to produce an expanded training instance. Based on the AUC measurements repeated in [56] SNNB and LWNB did not show to significantly improve the NB, and LBR performed worse than NB. According to experimental results, LNB was slightly better than NB and C4.4, in terms of accuracy, robustness and stability.

In another work, Irani *et al.* [83] focused on the social spam profiles in MySpace. Here they compared well known machine learning algorithms (AdaBoost algorithm, C4.5, SVM, NNs, NB) with respect to their abilities to distinguish spam profiles from legitimate profiles. According to the simulations on over 1.9 million MySpace profiles, C4.5 DT algorithm achieved the highest accuracy (99.4%) of finding the spam profiles while NB achieved 92.6% accuracy. Here each user was represented with a social network profile. Each profile included two kinds of data which are categorical data (i.e. sex, age, relationship status)

Table 3-2 Comparison of the clustering methods

Clustering Methods / Factors	Partitional Clustering	Hierarchical Clustering	Density-Based Clustering
Time and Space Complexity	Time Complexity $O(NQL)$ Space Complexity $O(Q+N)$	Time Complexity $O(N^2 \log N)$ Space Complexity $O(N^2)$	Time Complexity $O(N \log N)$ Space Complexity $O(N)$
Clustering Type	Used for summarization	Used for taxonomy	Used for finding core points
Cluster Type	Produces globular clusters Data objects are similar within the same cluster and dissimilar between different clusters (Prototype and graph based)	Single link and density-based clustering produce non-globular clusters	
Data Objects	Can work with different types of data objects and can handle few hundreds of clusters Works well with low or moderate dimension		Works well with moderate or high dimension
Data Set	Works well with different small and medium size data sets Requires proximity measures	Works well with different small and medium size data sets Requires proximity matrix	Works well with large datasets

and free-from data (text information i.e. about me, interests). Simulations were performed on Weikato Environment for Knowledge Analysis (WEKA) platform where classifiers' default settings were used with 10 fold-cross validation.

Although using classifiers for user profiling has been studied in the literature (as explained in Chapter 2), the related works (Subsection 2.2.4.) show that a limited number of classifiers have been investigated for user profiling. The simulations in Subsection 3.3.1., aim to compare the classifiers' performances with different user profile datasets.

Our previous works [84], [85] and [86] have been the first in the literature to present the comparison of the classification accuracy performance of different classification algorithms with user profiles. In [85] NB, IB1, BN and LBR classifiers were compared using a user profiling dataset. Furthermore in [86] tree-based algorithms to be used for user profiling (i.e. Classification and Regression Tree (SimpleCART), NBTree, Id3, J48 -a version of C4.5- and Sequential Minimal Optimization (SMO)) were included and compared with large user profile data. In the next section in more details the results taken from [85] and [86] will be discussed.

3.3.1. Simulations and Results

A. Dataset

Simulations were conducted using a variety of user profile datasets that reflect the users' personal information (demographic data), interests and preferences information.

Here, as a demographic profile data, UCI's adult dataset [80] has been modified and used. This dataset has been selected because;

- it is a real world dataset,
- it is an open access dataset, and therefore its utilization does not raise any data confidentiality issues which are essential for user profiling and personalization studies, and
- the information within this dataset is general and it can be classified and used as a demographic profile.

Before the simulations, attributes were normalized and discretized using unsupervised attribute filters.

B. Simulations

All simulations were performed in the WEKA machine learning platform providing a benchmarking consisting of a collection of popular existing learning schemes that can be used for practical data mining and machine learning applications [69]. There are over 250 publications [87] including conference papers, a thesis and a book [69] in which WEKA has been used and referenced.

This section is divided into two subsections. In the first subsection (see Subsection 3.3.1.1.) the results are obtained and presented by comparing the fundamental classification algorithms. In the second subsection (see Subsection 3.3.1.2.) DT methods together with SMO are analysed for user profiling. The SMO classifier implements the sequential minimal optimization algorithm for the training of a SVM classifier [69].

The key points for the simulations are highlighted as follows;

- Datasets have been converted into WEKA readable ".cvs" format (see Table 3-3) files. In this table missing values are indicated with the "?" symbol.

- In Subsection 3.3.1.1. two sets of simulations are carried out. The first simulation, simulation 1a, is conducted on a user profile dataset with 20 instances and 10 attributes (see Table 3-4). The second simulation, simulation 1b, on the other hand, is performed using the user profile dataset with 20 instances and 18 attributes. These attributes are Age, Work-class, Final-weight, Education, Education-num, Marital-status, Occupation, Relationship, Race, Sex, Native-country, Capital-gain, Capital-loss, Hours-per-week, Interest-music, interest-book, interest-sport and Preference-sound. Please note that in Table 3-3 only the demographic user profile is presented, including 20 instances and 10 attributes.

- As a test mode 10 fold cross-validation is chosen where 10 pairs of training sets and testing sets are created. All previously mentioned classification algorithms will be evaluated based on the same training sets and than tested on the same testing sets to obtain the classification accuracy.

3.3.1.1. Simulations I

In this subsection the results of four classifiers (NB, BN, LBR and IB1) for the selected user profile dataset are compared. The parameters for these simulations are carefully selected to demonstrate real application scenarios.

Assume the first 20 instances of the UCI's adult dataset are chosen, as shown in Table 3-3 [80].

Table 3-3 Personal user profile data in ".cvs" format

Age	Work-class	Education	Education-num	Marital status	Occupation	Relationship	Race	Sex	Native country
25	Private	11^{th}	7	Never-married	Machine-op-inspct	Own-child	Black	Male	United-states
38	Private	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	United-states
28	Local-gov	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	United-states
44	Private	Some-collage	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	United-states
18	?	Some-collage	10	Never-married	?	Own-child	White	Female	United-states
34	Private	10^{th}	6	Never-married	Other-service	Not-in-family	White	Male	United-states
29	?	Hs-grad	9	Never-married	?	Unmarried	Black	Male	United-states
63	Self-emp-not-inc	Prof-school	15	Married-civ-spouse	Prof-specialty	Husband	White	Male	United-states
24	Private	Some-collage	10	Never-married	Other-service	Unmarried	White	Female	United-states
55	Private	7^{th}-8^{th}	4	Married-civ-spouse	Craft-repair	Husband	White	Male	United-states
65	Private	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	United-states
36	Federal-gov	Bachelors	13	Married-civ-spouse	Adm-clerical	Husband	White	Male	United-states
26	Private	HS-grad	9	Never-married	Adm-clerical	Not-in-family	White	Female	United-states
58	?	HS-grad	9	Married-civ-spouse	?	Husband	White	Male	United-states
48	Private	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	United-states
43	Private	Masters	14	Married-civ-spouse	Exec-managerial	Husband	White	Male	United-states
20	State-gov	Some-collage	10	Never-married	Other-service	Own-child	White	Male	United-states
43	Private	HS-grad	9	Married-civ-spouse	Adm-clerical	Wife	White	Female	United-states
37	Private	HS-grad	9	Widowed	Machine-op-inspct	Unmarried	White	Female	United-states
40	Private	Doctorate	16	Married-civ-spouse	Prof-specialty	Husband	Asian-Pac.	Male	?

Table 3-4 demonstrates the classification accuracy results of these four classifiers obtained from the simulation 1a. It can be seen from Table 3-4, NB and IB1 classifiers have a classification accuracy of 95%, where 19 dataset instances have been classified correctly and 1 instance has been classified incorrectly. Moreover, the second best result is 90%, belonging to the LBR

classifier followed by NB and IB1 algorithms. The BN result is the worst at 85% (17 correctly classified and 3 incorrectly classified instances). Therefore, both NB and IB1 methods outperform the LBR and BN classifiers in terms of classification accuracy.

Table 3-5 shows that the precision of the four classification algorithms are very similar at about 0.95. Precision is one of the performance measures and differs from accuracy, it does not relate to the true value (accepted reference value) [77]. Precision shows the closeness of the independent test results on homogeneous data and usually computed as a standard deviation of the results [77]. As previously mentioned, test mode of the simulations in this section is 10 fold cross-validation where the dataset is partitioned into 10 subsets. One of the subsets was used as the training dataset and the other subset was used as the test dataset, and this process repeated 10 times, once for each subset that was used as the test dataset. Here the classification accuracy is the average of 10 runs and precision is the standard deviation of the random errors from each run.

Figure 3-11 shows the error rate results. Here four different parameters are used to represent the error rate of the four classification algorithms. These are; Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Relative Absolute Error (RAE) and Root Relative Squared Error (RRSE).

This figure shows that NB and IB1 classifiers have the lowest error rates. Furthermore, the BN classifier has the highest error rate and the difference is higher in RRSE and RAE. Based on the above classification accuracy results (see Table 3-4), the BN classifier demonstrates the highest error rate (see Figure 3-11).

Table 3-4 Classification accuracy test results (simulation 1a)

Classifier	Correctly classified instances	Incorrectly classified instances
NB	19 (95%)	1 (5%)
IB1	19 (95%)	1 (5%)
LBR	18 (90%)	2 (10%)
BN	17 (85%)	3 (15%)

Table 3-5 Classifiers vs. precision

Classifier	Precision
NB	0.95
IB1	0.95
LBR	0.947
BN	0.944

In order to compare the classification accuracy performance of the NB, BN, LBR and IB1 classifiers with the user profile data, a second simulation, simulation 1b, was performed on the extended user profile dataset. During the second set of simulation the following results were obtained;

- The classification accuracy performance of the BN classifier is 80%. Therefore, when this result is compared with the simulation 1a it can be seen that BN classifier's performance decreases 5% from 85% to 80%.

On the other hand, for NB, IB1 and LBR classifiers, simulation 1a results have remained the same during the simulation 1b (see Table 3-6). Therefore, NB and IB1 classification algorithms keep performing well with a bigger user profile dataset.

It is known that a LBR classifier was proposed to improve the performance of NB classifier by applying the lazy algorithm on the NB classifier and reducing the conditional independence assumption of the NB. According to the simulation results, NB outperforms both BN and LBR classifiers. This is due to the fact that the NB classifier assumes that class attributes within the same class are conditionally independent given the class label and the attributes within the used user profile dataset are independent from each other.

- Figure 3-12 shows the error rate results of the four classifiers. According to these results, in the second simulations (simulation 1b) RAE of LBR and BN classifiers have increased significantly. This increment is significantly higher in the BN classifier where RAE increases from 121% to 162%.

Unlike the three previously discussed algorithms, LBR cannot handle numeric attributes. Therefore, before simulations with LBR, the attribute values of both datasets were normalized using unsupervised attribute filters “Normalized” and “Numeric-To-Binary”.

3.3.1.2. Simulations II

In this second part of the simulations section, the results of seven classifiers (NB, IB1, SimpleCART, NBTree, Id3, J48 and SMO) are compared.

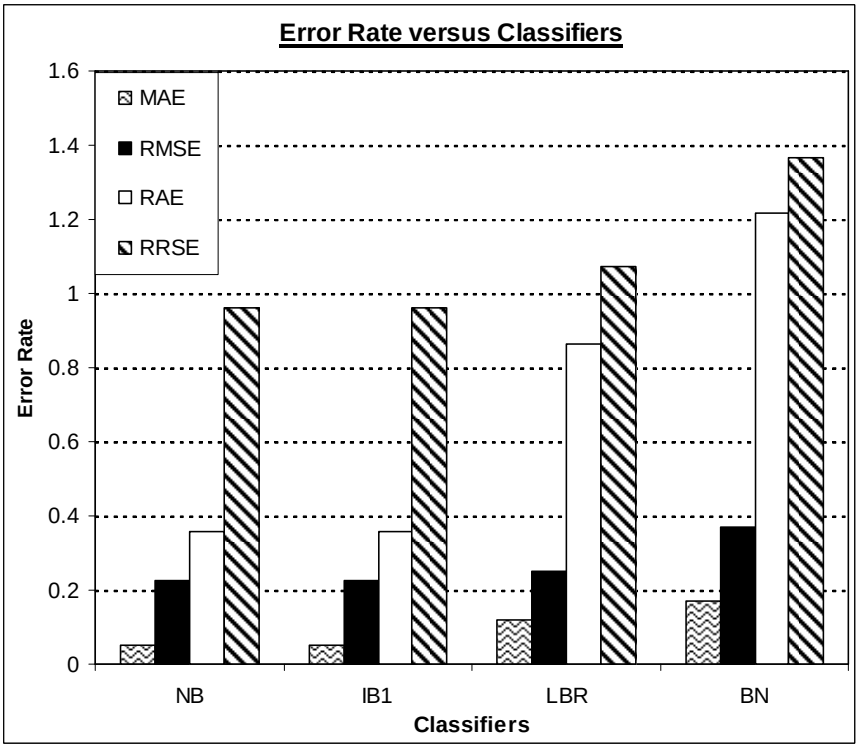


Figure 3-11 Error rate measures of classifiers (simulation 1a)

Table 3-6 Classification accuracy test results (simulation 1b)

Classifier	Correctly classified instances	Incorrectly classified instances
NB	19 (95%)	1 (5%)
IB1	19 (95%)	1 (5%)
LBR	18 (90%)	2 (10%)
BN	16 (80%)	4 (20%)

Four different user profile datasets have been used for the simulations (see Table 3-7). Each dataset has the same number of attributes and different number of instances, varying from 150 to 1000 instances. Here the focus is on the simulations conducted on the user profile dataset D. In Table 3-8 a comparison of the results is done with respect to the classification accuracy (2nd

column) and the time taken to build the model (3rd column). The time taken to build the model is the system time that was used to run the classifier and is converted from millisecond into seconds by WEKA.

According to the results the NBTree classifier performed better than all other classifiers with a classification accuracy of 90.20% (see Table 3-8). Here, the NBTree classifier classified 902 instances correctly out of 1000. The J48 classifier follows the outcome of NBTree classifier with the second highest result which is 89.90%. Consequently, the SimpleCART shows a performance of 89.50% where 895 instances classified correctly out of 1000 instances.

According to the Table 3-8, Id3 classifier gives the worst result of 74.30%.

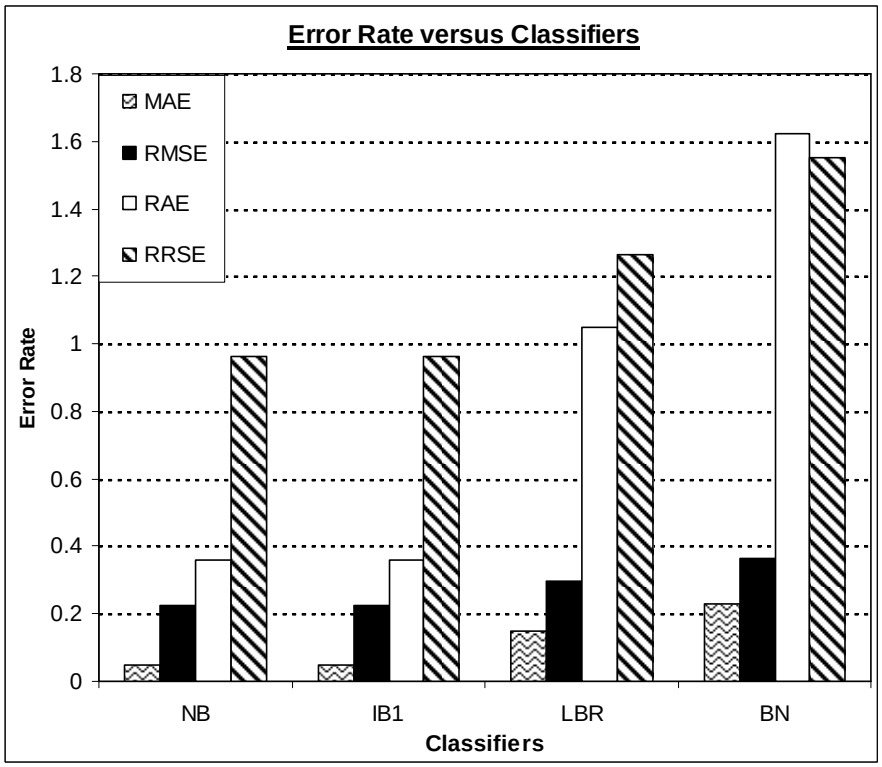


Figure 3-12 Error rate measures of classifiers (simulation 1b)

Table 3-7 User profile datasets

User Profile Dataset Name	Number of Attributes	Number of Instances
Dataset A	18	150
Dataset B	18	327
Dataset C	18	746
Dataset D	18	1000

Table 3-8 Classification accuracy performance of the classifiers along with time taken to build the model

User Profile Dataset / Algorithms	Dataset D (18 attributes 1000 instances)	
NBTree	90 20%	19 67 sec
SimpleCart	89 50%	6 75 sec
J48	89 80%	0 05 sec
NB	88 90%	0 sec
SMC	88 90%	34 27 sec
IB1	82 90%	0 sec
Id3	74 30%	0 11 sec

It can be observed from the above analysis that NBTree classifier gives the best classification accuracy results. Moreover, SimpleCART and J48 classifiers give very similar results to NBTree. The J48 is the enhanced version of the C4.5 classifier and has been developed to address the problems of both C4.5 and Id3 classifiers. Therefore, it was expected from the J48 classifier to have better

classification accuracy performance than Id3 and this was confirmed from the results.

In our previous work [85], it was found that, on a very limited user profile dataset, NB and IB1 classifiers have the same classification accuracy results. However, from this study it can be observed that NB classifier results in a better classification accuracy than the IB1 classifier for a relatively large user profile dataset.

It can be seen from the Table 3-8 that the SMO classifier has the highest time requirement to build the model in all simulations. Furthermore, with the second highest time requirement, NBTree followed the outcomes of SMO. It is also noticeable that SimpleCART classifier has the third highest time requirement in all simulations.

Although J48, Id3, NB and IB1 classifiers need less time to build the model, as far as the lowest time requirement is concerned IB1 and NB seem to be the most relevant classifiers.

It is clear from the above results that NBTree classifier has the best classification accuracy performance but with one of the highest time requirements. The NBTree classifier is a hybrid classifier that generates DT with NB at the leaves node and obtains the advantages of both classifiers. Therefore it is reasonable that NBTree achieves better classification accuracy than the NB classifier and DT classifiers (i.e. Id3, J48, SimpleCART). Moreover, this integration comes with the complexity which results in one of the highest time requirements to build the classification model.

3.4. Discussions

Table 3-9 summarizes the popular classifiers and compares their characteristics including the findings from the previous section. It can be seen from the table that each algorithm has different performances on different domains (i.e. ranking, spam profile detection). DTs are complex structured algorithms (see Subsection 3.1.2.) and with large datasets they can be computationally expensive [88]. In user profiling DTs gave good classification accuracy (see Subsection 3.3.1.2.). However, the dataset used for simulations was relatively large. In user profiling applications with large datasets, using DT algorithms to classify will not be feasible in terms of time and space requirements. BN and NB classifiers both rely on the Bayes' rule. BN classifier represents uncertain attribute dependencies whereas NB assumes conditional independency. Compared to the DT algorithms, NB is simple and computationally efficient. With fast training, test data analysis and decision making, the NB algorithm performed very similar to the DT algorithms in terms of classification accuracy.

Hence, it can be argued that in user profiling, NB is a better option compared to DT algorithms. However, because of the conditional independence assumption, NB can lead to incorrect probability estimations that can reduce the correct classification accuracy. User profiles have a semi-static/dynamic structure that includes both numeric and symbolic attributes. However, not each attribute is independent from other attribute/attributes (e.g. user's age can effect the favourite music type). Hence, an increase in the number of user profiles and/or related attributes can decrease the NB's performance. For example, this decrease can be observed from the Table 3-4 and Table 3-8. LBR is a lazy

learner that relaxes the conditional independence assumption of NB by applying a lazy algorithm. The LBR classifier is effective with small datasets [57] hence, similar to NB, it may not perform well with large user profile datasets.

IBL also performs well with the user profile dataset. This algorithm assumes that similar instances have similar classifications [89]. Similarly, in user profiling, users with similar profiles are likely to share similar personal interest and preferences. However, performance of this algorithm degrades on the presence of irrelevant attributes which can be the case in user profiles. The success of SMO in text classification and pattern recognition can also be observed in user profiling as well. However, this can be an expensive and time consuming option for large user profile datasets.

To the best of the author's knowledge, the study carried out in this chapter is the first in the literature to;

- Investigate various classification and clustering algorithms for the user profiling and evaluate their performances with different user profile datasets.

From the given information, simulation results and comparisons of the algorithms, the utilization of the IBL algorithm for user profiling is focused. This is because, compared to the other algorithms, IBL has the following properties;

- processes instances incrementally,
- is fast and robust,

Table 3-9 Comparison of the most popular classifiers

Classifiers	Algorithms	Works	Dataset	Performance	Advantages/Disadvantages
Bayesian Classifiers	NB	[78] Panda et al [79] Zang and Su [82] Hung and Ling [83] Farid et al	KDDCup 99 dataset 15 UCI dataset 3 artificial 18 UCI dataset 1.9 million myspace social profiles	Performs well in ranking problems finding spam profiles network intrusion detection and have similar classification accuracy with SVM on UCI datasets	Simple and computationally efficient However conditional independence assumption may lead to incorrect probability estimations
	BN	[59] Pena et al [85] Cufoglu et al	2 UCI dataset 4 synthetic dataset User profile dataset	Performs well with UCI dataset and has average performance with user profile dataset	It can represent uncertain attribute dependencies However it has high computational complexity and needs complete knowledge of prior and conditional probabilities
Lazy Learners	LBR	[56] Jiang and Guo [57] Weng et al	32 UCI dataset 36 UCI dataset	Performs well with small dataset	Relaxes the conditional independence assumption of NB by applying lazy algorithm on the NB However it does not performs better than NB in every domain and it is computationally efficient with small datasets
	IBL	[68] Aha et al	5 UCI dataset	Performs well with real world datasets that have noise and irrelevant attributes	It processes instances incrementally It is fast robust and can represent probabilistic and overlapping concepts However each attribute has the same influence on classification regardless their relevancy
Decision Trees	J48	[56] Jiang and Guo [82] Hung and Ling [83] Farid et al	3 artificial 18 UCI dataset KDDCup 99 dataset 1.9 million myspace social profiles	Performs well in finding the spam profiles and robust in detecting new intrusions	It is an enhanced version of C4.5 and has been developed to address the problems of both C4.5 and id3
	Id3	[78] Panda et al	KDDCup 99 dataset	Performs well in detecting new intrusions	It can handle nominal attributes and nominal classes It can also deal with missing class values and data that contains no instances However It cannot handle numeric attributes and missing attribute values
Support Vector Machine	SVM	[82] Hung and Ling [83] Farid et al	3 artificial 18 UCI dataset 1.9 million myspace social profiles	Performs average in finding spam profiles and have similar classification accuracy with NB on UCI datasets	Complex and computationally expensive However it has high performance in practical applications such as text classification and pattern recegnition

- can represent probabilistic and overlapping concepts,
- assumes that the similar instances have similar classification that is similar to the concept of the user profiling where similar users with similar profiles share similar personal interest and preferences, and
- has potential to be improved to give better performance for user profiling.

However, similar to other algorithms mentioned in this chapter, IBL does not consider the relevancy of the user profile information during the user profiling which is an important factor in achieving accurate user profiles. For this purpose, feature weighting can be introduced to improve IBL. To the best of the author's knowledge, this research is the first work to adapt the IBL for user profiling and modifies it to carry out feature weighting to classify user profiles.

3.5. Summary

This chapter described classification and clustering algorithms for user profiling. For this purpose, the characteristics of both classification and clustering have been presented. In addition, the classification algorithms, which are Decision Trees (DTs), Nearest Neighbour (NN) Classifiers, Support Vector Machine (SVM) and Bayesian Classification, were described in detail. The most popular clustering methods were also discussed. The clustering methods presented in this chapter were: Hierarchical clustering, Partitional clustering and Density-based clustering. A comparison of these methods was carried out, addressing the time and space complexity, clustering type, cluster type, data objects and data set factors of each clustering method. The research works carried out with the classification algorithms were described. Following this, classification

accuracy performances of the better known classifiers such as BN, NB and NBTree were simulated using the user profile data with the results presented.

The classification and clustering algorithms, the related works and the simulations that have been discussed and presented in this chapter does not consider the relevancy of the user profile information during the user profiling. Relevancy of the information is an important factor to achieve accurate user profiles. In Chapter 4 the feature weighting methods, which balances the effect of relevant and irrelevant user information during classification, will be discussed.

Chapter 4

Existing Weighting Methods

A number of feature (attribute) weighting methods have been proposed to reduce the impact of the irrelevant and weakly relevant features as well as to increase the impact of the strongly relevant features when calculating distance measure between instances [90]. The relevant features are indispensable since their absence will cause the loss of prediction accuracy [91]. Furthermore, the weakly relevant features can sometimes contribute to the prediction accuracy but irrelevant features do not contribute [91].

Some of these works attempt to categorize the existing feature weighting methods [92][93]. In particular, Wettschereck et al. [92] proposed a five-dimensional framework to categorize the automated weight-learning methods. In this chapter the feature weighting methods based on the first dimension only are investigated.

The first dimension is the feedback dimension (see Figure 4-1) [92]. This dimension concerns whether or not the feature weighting method receives

feedback from the classification algorithm [92]. Here, the feature weighting methods with the feedback are known as 'feedback methods (wrapper methods)' and the methods without the feedback are known as 'Ignorant methods (Filter methods)'.

In this chapter both wrapper methods and filter methods are described in detail. Section 4.1. focuses on the filter models while Section 4.2. describes wrapper methods. Section 4.3. discusses the filter methods and the wrapper methods for user profiling. Finally, a summary is given in Section 4.4. All of the equations within this chapter were written based on the assumptions in Subsection 2.1.2.3.

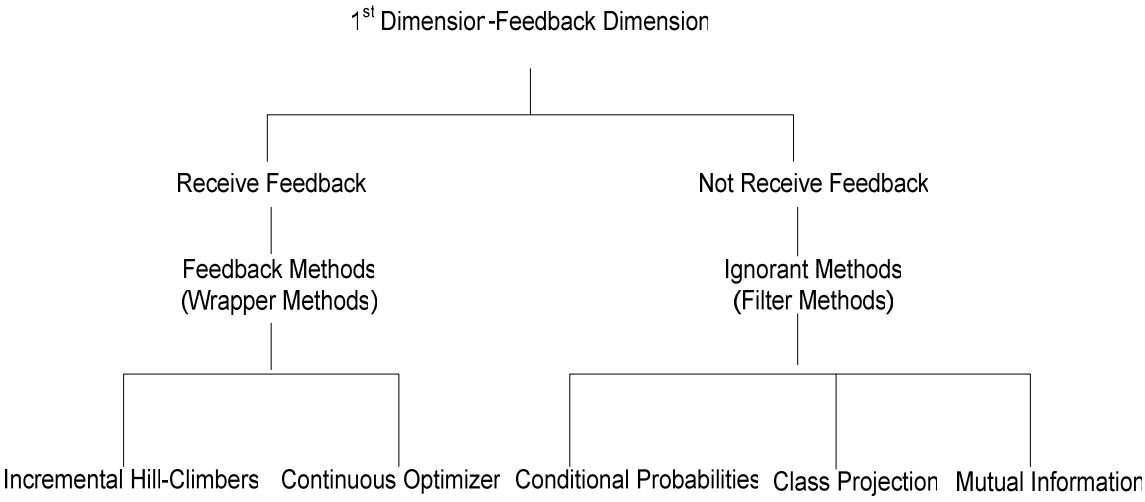


Figure 4-1 Feature weighting methods

4.1. Filter Methods

Filter methods use only the training data to calculate and assign the feature weights [94]. These methods are independent from the classifier's feedback and as a result are much faster than the wrapper methods [95]. Therefore, filter methods are considered to be effective and efficient to suit the data sets with

large dimensions. The main drawback of the filter methods is that they totally ignore the effect of the selected feature subset on the performance of the classifier. Hence, these methods cannot efficiently filter the redundant or even harmful features for generalization [96]. Relief [97] and FOCUS [91][98] are two of the existing algorithms that fall into the filter methods.

Relief is a feature weighting algorithm proposed by the IBL [97]. Given training data, Relief detects and assigns relevant weight to those features which are statistically relevant to the target concept (label value) [97]. Relief is a randomized algorithm as it samples training set instances randomly and updates the feature weight based on the difference between the selected instance and the two nearest instances of the same (the 'near hit') and opposite (the 'near miss') class [95][97][98].

The FOCUS algorithm exhaustively examines all the features and finds the minimal set of features (min-features) that are sufficient to determine the concept for all instances in the training set [91][98]. Given enough training data, FOCUS will select none of the irrelevant features, all of the strongly relevant features and the smallest subset of weakly relevant features which are sufficient to determine the concept [91].

Figure 4-2 [98] shows the view of the feature relevancy of Relief and FOCUS. It can be seen that FOCUS is searching for a min-feature while Relief searches both weakly and strongly relevant features.

Three main filter methods are; Conditional Probabilities, Class Projection, and Mutual Information.

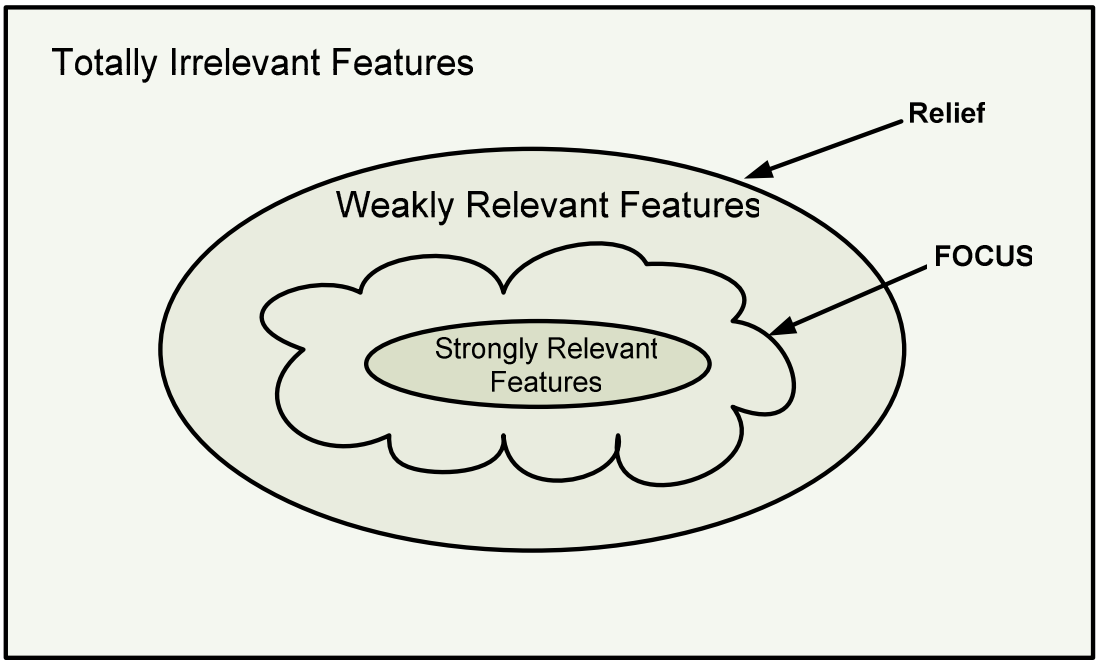


Figure 4-2 Relief vs. FOCUS [98]

4.1.1. Conditional Probabilities

Conditional probabilities filter methods weight the features based on the correlation. Two of the well known conditional probability feature weight methods are, Per-Category Feature (PCF) importance and Cross-Category Feature (CCF) importance. These filter methods assign weights of the features by using conditional probabilities [92].

In PCF, same feature is assigned with different weights for each category that the feature is found in [99]. Here, the PCF calculates the conditional probability for each feature in every category and assigns high weight values to features having high correlation within the given category [92]. As a result, the importance of a feature is different in different categories. The feature weight calculation formula of PCF is as follows [99]:

$$w(f_k) = P(C_m \mid f_k) \tag{4-1}$$

In contrast to the PCF, the CCF assigns the same weight for the same feature in each category the feature is found. The CCF calculates the conditional probability for each feature in every category and takes the sum of the squares of these conditional probabilities to find the weight of the feature. Here, the importance of the feature is the same in different categories. The following formula is used for the feature weight calculation [99];

$$w(f_k) = \sum_{m=1}^{Q} P(C_m \mid f_k)^2 \tag{4-2}$$

4.1.2. Class Projection

Class projection filter method weights the features based on the distribution. Here, features are assigned higher weights if the distributions of the feature values across the classes are highly skewed [92].

Value Difference Matrix (VDM) is a popular class projection method. In VDM, the feature has the same weight for different categories where the feature is found. During the feature weight calculation, the conditional probability calculation is performed based on the feature value [100]. Therefore, this group of filter methods assign feature weights based on the feature's value. Here VDM finds the frequency of the various values of the feature, squares them, sums them and finally takes the square root of the result to compute the weight thus [100]:

$$w(f_k) = \sqrt{\sum_{m=1}^{Q} P(C_m \mid f_k(v_k))^2} \quad (4\text{-}3)$$

4.1.3. Mutual Information

Mutual Information (MI) is an information-theoretic measure of association between two words [101]. The MI between class C_m and feature f_k is defined as follows [101]:

$$MI(f_k, C_m) = \log \frac{P(f_k, C_m)}{P(f_k)P(C_m)} \quad (4\text{-}4)$$

This filter method assigns the feature weights using the MI between the feature's values and the class of the training examples as follows [92]:

$$w(f_k) = \sum_{f_k \in f} \sum_{C_m \in C} P(f_k, C_m).\log \frac{P(f_k, C_m)}{P(f_k)P(C_m)} \quad (4\text{-}6)$$

4.2. Wrapper Methods

Wrapper methods are feature weighting methods that use the classifier's feedback to guide the search to find the relevant attributes [94]. Hence, these methods take the biases of the classifier into account to explore and evaluate the optimal feature subset for the classification [95]. The use of these methods with a high-dimensional data set is costly and time consuming [94][95][96]. Two of the well known wrapper methods are, Incremental hill-climbers and continuous optimizer.

4.2.1. Incremental Hill Climbers and Continues Optimizers

These methods modify feature weights incrementally, to increase the similarity between a test instance and nearby training examples in the same class/category and decrease its similarity with nearby training instances in other categories [90][92]. This group of wrapper methods iteratively update feature weights using only the randomly selected training instances [90][92].

IB4 [102] and EACH [103] are two of the more well known wrapper algorithms. IB4 [102] is an incremental algorithm which has an incremental feature weighting function. Here, feature weights are increased when they correctly predict the class [104]. Moreover, incorrect prediction decreases the value of feature weights [104]. Similarly in EACH, each correct classification results in an increase in the weight where mismatch decreases the weight by the same amount (Δ) [92]. Here, the weight of the feature f_k is calculated as follows [105]:

$$w(f_k) = w(f_k) \pm \Delta \tag{4-7}$$

In this algorithm, for incorrect classification, the weights for the matching features are decremented while the weights for the mismatching features are incremented [92]. In IB4, weights are calculated for each feature and class label [104]. This algorithm can handle both numeric and symbolic attributes where the distance between symbolic attribute values is the Hamming distance [102].

4.3. Discussions

The use of the feature weighting methods for clustering can improve the accuracy of the classification process. Several studies in the literature have presented noteworthy improvements in the classification performance when these weighting methods are used [90] [93] [99]. However, the concept of feature weighting should be considered separately from the other studies when user profiling is intended. In Chapter 5 we will discuss and further propose weighting methods for the user profiling. This will be carried out through simulations and mathematical analysis. In addition, the use of “filter methods” on the proposed algorithm will be discussed in the next chapter.

In order to make use of the weight update equation (4-7) of “wrapper methods”, the system should be aware of whether a correct classification occurs or not. This information is used by the wrapper methods to increment or decrement the weight of each feature to achieve better classification accuracy. To enable such a weight assignment, either a training dataset, where the correct class information is already available, has to be used or the system needs to be informed on the correct class information after each decision is made. For the user profiling the latter case is not possible until the user provides feedback to the system. This would only benefit the system performance once the user is involved in the classification process. If wrapper methods are used over a training dataset, the training instances can be fed into the classifier and the weights can be updated comparing the output of the classifier with the training data itself. Genetic Algorithms and Neural Network are good candidates for this type of classification. However, it should also be noted that these methods are

costly and time consuming as the number of dimensions of the dataset increases.

The incremental hill climbers and continues optimizers perform iterative estimation and assignment of the feature weights. An iterative method may ease the computational complexity of the used algorithm by enabling the continuous update of the estimated weights without the need of a computationally expensive equation for each update. Therefore, filter methods, similar to equation (4-2), can be modified to simplify the process of weight update.

4.4. Summary

This chapter described two popular feature weighting methods, namely filter and wrapper methods. Filter methods calculate directly the weights of the features, while the wrapper methods calculate these weights iteratively. The characteristics, advantages and disadvantages of both methods were briefly discussed. Some well known filter and wrapper methods were identified in this chapter, such as conditional probabilities and Incremental hill-climbers. Finally, utilization of filter and wrapper methods for user profiling was discussed.

The next chapter, Chapter 5, will provide detailed information about the proposed clustering algorithm and the feature weighting algorithms for user profiling.

Chapter 5

Proposed Multi-Dimensional Clustering

This chapter describes a novel clustering algorithm and feature weighting method which is proposed to evaluate the importance of each feature and/or feature value for a better clustering performance for user profiling.

More precise clustering of the users can be achieved by increasing the number of features in the user profile, and therefore more detailed knowledge about a user's preferences, interests and needs can be obtained. However, not every feature contributes to the clustering accuracy the same way. Some features may be highly relevant to the clustering criterion and some may be quite irrelevant. A two-step methodology has been proposed, where

- *in the first step the relevance of each feature and/or feature value is assessed and then,*
- *in the second-step the clustering is performed by making use of this assessment,*

5.1. Instance Based Learner Algorithm for User Profiling

The Instance Based Learner (IBL) algorithm was presented in Chapter 3 in some detail. As far as user profiling is concerned IBL seems to be an appropriate methodology for classifying the user information. If IBL is used for user profiling, then it simply assigns the new user to the class that is associated with an existing user (training instance) who is closer to the new-comer (test instance) in terms of feature-by-feature comparison. IBL is a suitable algorithm for user profiling as users with similar profiles are likely to share similar personal interests and preferences.

A drawback of IBL is that it treats all the features the same regardless of their relevance. For instance, assume a profiling scenario where all users' features are nominal (or in other words none of them are numeric) for the sake of simplicity in understanding the scenario. There are A features and each may take v possible values. The user which is to be clustered, the "new-comer", can immediately be clustered, if and only if in the user database there is an exact match where it's all A nominal values are equal to that of the new-comer. Otherwise, if all except 1 feature are equal then there might be $v-1$ possible users in the user profile database that the new-comer can be matched with. In this case, the classification is not straightforward and it lies in the hands of some "supporting rules" to pick one of the $v-1$ users. A simple example for a supporting rule is to pick the user which comes first in the comparison process. Note that the scenario can get more complicated to handle as the number of

non-matching features increases, i.e. if α values are not matching there are $(v-1)^{\alpha}$ possibilities.

Having one or more numeric valued features in the profile may let the decision to be made easier over $(v-1)^{\alpha}$ values. Obviously this does not mean that the right class is assigned to the new-comer for the given scenario if the numeric features are present.

Please also note that the user profiles are usually composed of nominal values rather than numeric values, such as the personal interests like sports, music and books, or demographic information like nationality, level of education and occupation. Therefore, for user profiling applications the given scenario is considered to be realistic.

In the following section the author proposes the Multi-Dimensional Clustering (MDC) algorithm which modifies the IBL for improving the accuracy of the user profiling.

5.2. Multi-Dimensional Clustering Algorithm

In contrast to IBL, the proposed MDC assigns weights to the features and considers the weighted distance of the instances for clustering. Here, relevant features are aimed to have more influence on the clustering than irrelevant features. Three weighting methods for the MDC will be presented aiming to improve the performance.

In the proposed methods the distance function in (3-6) was modified as;

$$dist(X_{i,}Y_j) = \sqrt{\sum_{k=1}^{A} w_{k,l}\, g(x_i(k), y_j(k))} \tag{5-1}$$

where $w_{k,l}$ is the weight corresponding to the lth feature value of the kth feature. l is equal to the value of the $x_i(k)$. Therefore, the selection of which weight is to be used for a particular feature value is based on k and the $x_i(k)$. There are N_w weights, where N_w is equal to the number of feature values i.e. $N_w = \sum_{k=1}^{A} v_k$. Note that $g(x_i(k), y_j(k))$ is evaluated as it is in the original IBL algorithm.

In (5-1) the weight matrix W, composed of the $w_{k,l}$ values must be calculated. The W matrix is as follows;

$$W = \begin{bmatrix} w_{f_1,f_1(1)} & w_{f_1,f_1(2)} & \cdots & w_{f_1,f_1(v_1)} & 0_{1x(m-v_1)} \\ w_{f_2,f_2(1)} & w_{f_2,f_2(2)} & \cdots & w_{f_2,f_2(v_2)} & 0_{1x(m-v_2)} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ w_{f_A,f_A(1)} & \cdots & \cdots & w_{f_A,f_A(v_A)} & 0_{1x(m-v_A)} \end{bmatrix} \tag{5-2}$$

As can be seen, W is a matrix of size A rows by m columns, where $m = \max(v_k)$ and $\max(\bullet)$ represents a function which picks the largest of its corresponding term for $k = 1,2,3,...,A$. Notation 0_{1xb} defines an all zero vector *of* size $1xb$. 0_{1xb} simply fills W with zeros where the number of feature (attribute) values in a row is less than m.

In this method W should be designed to be “dynamic”. Thus, after the arrival of each user, the weighting values $w_{k,l}$ and the weight matrix have to be updated.

In the next three subsections three different methods for the calculation of $w_{k,l}$, and therefore for the formation of W, will be introduced.

5.2.1. MDC weight method by Cross Clustering (MDC-CC)

Several feature weighting methods have been proposed in the literature (see Chapter 4). These methods aim to reduce the impact of the irrelevant and weakly relevant features and to increase the impact of the strongly relevant features when calculating distance measure [90]. It is well known that the wrapper methods are costly and time consuming with high-dimensional data, while filter methods are effective and efficient to suit such data types. Hence, considering the multi-dimensionality of the user profile data, this research is focused on the filter methods. According to Wettshereck *et al.* [92], the Cross Category Feature (CCF) method is one of the filter feature weight methods, which uses conditional probability to assign weights to the features. The CCF method assigns the same weight for the same feature on each category the feature is found in. Therefore, the importance of the feature is the same in different categories.

In the new proposed method, the CCF formula has been modified as follows to calculate the weights of the feature values for the MDC;

$$w_{k,l} = \sum_{m=1}^{Q} P(C_m \mid f_k(l))^2 \tag{5-3}$$

In (5-3) $P(C_m \mid f_k(l))$ represents the probability density function (pdf) of the mth cluster (C_m), given the lth feature value of the kth feature ($f_k(l)$). The

assignment of the feature weights according to the given criterion in (5-3) will be called MDC weight method by Cross Clustering (MDC-CC).

In addition, the following analysis on the main features of the MDC-CC weighted clustering algorithm for user profiling is proposed by author;

- For a given feature value, if the clusters are equally distributed then the $w_{k,l}$ obtains its lowest value i.e. $\frac{1}{Q}$. This means that $f_k(l)$ is not very useful for clustering the test instance in case of equi-probability where the feature value is uniformly distributed across all clusters.

- For a given feature value, if the probabilistic distribution of the clusters becomes uneven then the $w_{k,l}$ increases. This means that if $f_k(l)$ is not very likely to occur in each cluster then it is very useful during the clustering. Therefore, $w_{k,l}$ gets its maximum value, i.e. 1 when a feature value is perfectly correlated with one cluster.

Proof: If the clusters are equally distributed for the feature value $f_k(l)$ then;

$$w_{k,l} = \sum_{m=1}^{Q} (\frac{1}{Q})^2 + (\frac{1}{Q})^2 + ... + (\frac{1}{Q})^2 = \alpha \qquad (5\text{-}4)$$

Assume that the equi-probability is destroyed by changing the probability of one of the clusters by β and,

$$\beta = \beta_1 + \beta_2 + ... + \beta_{Q-1} \quad \text{, where} \quad \beta_1, \beta_2, ..., \beta_{Q-1} \in \mathrm{Q}^+ \qquad (5\text{-}5)$$

Then,

$$w_{k,l} = (\frac{1}{Q} + \beta)^2 + (\frac{1}{Q} - \beta_1)^2 + (\frac{1}{Q} - \beta_2)^2 + ... + (\frac{1}{Q} - \beta_{Q-1})^2$$

$$... = (\frac{1}{Q^2} + \frac{2\beta}{Q} + \beta^2) + (\frac{1}{Q^2} - \frac{2\beta_1}{Q} + \beta_1^2) + (\frac{1}{Q^2} - \frac{2\beta_2}{Q} + \beta_2^2) + ... + (\frac{1}{Q^2} - \frac{2\beta_{Q-1}}{Q} + \beta_{Q-1}^2)$$

$$... = (\frac{1}{Q^2} + \frac{1}{Q^2} + ... + \frac{1}{Q^2}) + (\beta^2 + \beta_1^2 + \beta_2^2 + ... + \beta_{Q-1}^2) > \alpha$$

Based on above calculations it is concluded that the minimum weight α is assigned for the equi-probable clusters and any change in the probability of a cluster increases the corresponding feature's weight values.

Since the sum of the probability distribution of the features across the clusters is equal to *1, i.e.* $P_1 + P_2 + ... + P_Q = 1,$ and each of the probability distribution is less than *1, i.e.* $0 < P_m < 1,$ then;

$$(P_m)^2 < P_m < 1 \text{ , therefore, } \quad (P_1)^2 + (P_2)^2 + ... + (P_Q)^2 < 1 \tag{5-6}$$

(5-6) is always true except for a cluster t where,

$$P_m = \begin{cases} 1, m = t \\ 0, m \neq t \end{cases} \tag{5-7}$$

The weight of a feature, the 'pdf' which satisfies (5-7), is $w_{k,l} = 1$.

Based on the above arguments the minimum and maximum values that $w_{k,l}$ can get is $\frac{1}{Q} \leq w_{k,l} \leq 1.$

Another way of assigning the weights of the feature values for user profiling could be through Per Category Feature (PCF) weighting (see Subsection 4.1.1.). For PCF, (5-3) should be written as follows,

$$w_{k,l}(C_m) = P(C_m \mid f_k(l)). \tag{5-8}$$

Here the weight values are shown by $w_{k,l}(C_m)$. This indicates that, different from the CCF, in PCF there are a several number of weights for each feature value. In (5-3) and (5-8) it is shown that there are $N_w = \sum_{k=1}^{A} v_k$ number of weights to be used along with the CCF where in PCF this number is $N_w = \sum_{m=1}^{Q}\sum_{k=1}^{A} v_k$. Because of this increase on number of weights, in PCF the weight matrix W is three dimensional and represented as follows,

$$W = \left[W(C_1);\quad W(C_2);\quad \cdots\quad W(C_Q)\right]$$

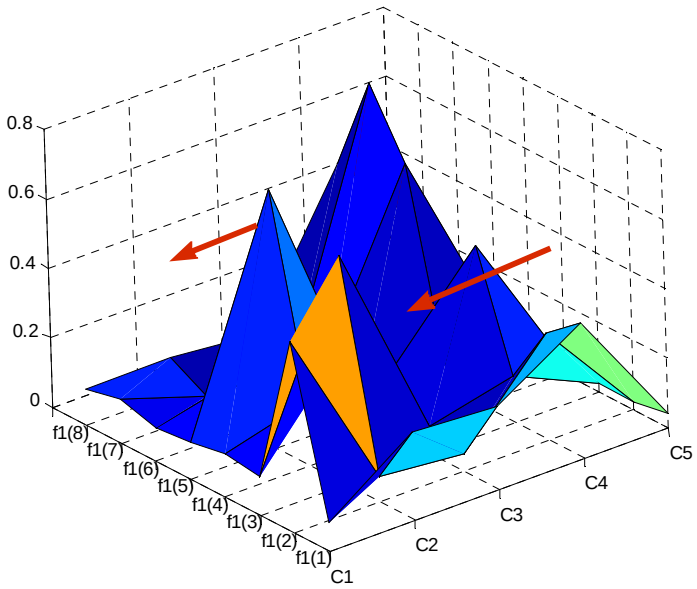
where

$$W(C_m) = \begin{bmatrix} w_{f_1,f_1(1)}(C_m) & \cdots & w_{f_1,f_1(v_1)}(C_m) & 0_{1x(m-v_1)} \\ \vdots & \ddots & \vdots & \vdots \\ w_{f_A,f_A(1)}(C_m) & \cdots & w_{f_A,f_A(v_A)}(C_m) & 0_{1x(m-v_A)} \end{bmatrix} \tag{5-9}$$

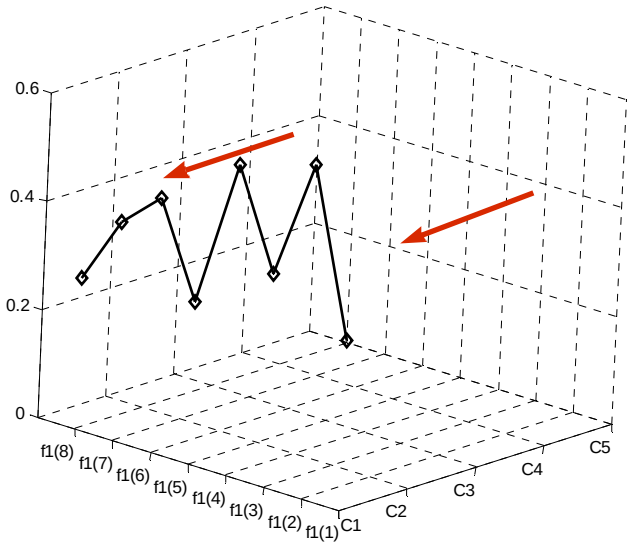
The below Figure 5-1(a) shows the three dimensional weight matrix for PCF. Here, if the square of the probabilities is summed up in the direction that the red-arrow points in Figure 5-1 (a), the CCF weights are obtained (see Figure 5-1(b)).

Implementation of PCF for IBL: The direct use of PCF with IBL for user profiling is not possible. This is due to the fact that the correct clustering probability of PCF is lower than the probability of incorrect clustering if IBL is used.

Proof: The weight values $w_{k,l}(C_m)$ are utilized within the clustering if the corresponding feature value is $x_i(k) \neq y_j(k)$, as shown in (3-5).



(a)



(b)

Figure 5-1(a) Representation of 3-dimensional weight matrix for PCF (b) Representation of 2-dimensional weight matrix for CCF, where the CCF weights are obtained by summing up the squares of each element in the direction that the red arrows show

The i th test instance and the two training instances, j th and $(j+1)$ th training instances, are in the following form; $X_i, Y_j \in C_m$ and $Y_{j+1} \in C_{m+1}$. Assume that

$x_i(k) \neq y_j(k)$ and $x_i(k) \neq y_{j+1}(k)$, and therefore the weights are incorporated into the clustering process. There are two possibilities:

Possibility#1: If $P(C_m \mid f_k(l)) > P(C_{m+1} \mid f_k(l))$, then according to (5-8) $w_{k,l}(C_m) > w_{k,l}(C_{m+1})$. This means that, because of the search for the minimum distance, X_i is more likely to be clustered in cluster C_{m+1}. Hence, test instance X_i will end up in incorrect cluster.

Possibility#2: If $P(C_m \mid f_k(l)) < P(C_{m+1} \mid f_k(l))$, then $w_{k,l}(C_m) < w_{k,l}(C_{m+1})$, which means that X_i is more likely to be clustered in C_m, which will give the right answer.

Here we skip the case where $P(C_m \mid f_k(l)) = P(C_{m+1} \mid f_k(l))$, as it is less likely to occur and will cause ambiguity.

According to the two possibilities previously given, the correct clustering could only be done if $P(C_m \mid f_k(l)) < P(C_{m+1} \mid f_k(l))$. As it was assumed at the beginning of this proof $X_i \in C_m$, so for PCF method it can be observed that the probability of correct clustering is less then incorrect clustering.

5.2.2. MDC weight method by Balanced Clustering (MDC-BC)

The weighting method, given in (5-3), depends solely on the conditional probability $P(C_m \mid f_k(l))$. In other words MDC-CC considers only a single parameter for evaluating the participation performance of each feature value in the clustering process. In this section other parameters in the process of weight calculation will be used in order to evaluate the relativity of each feature to the

clustering process. Therefore probability distribution of the clusters independent from the feature values, i.e. $P(C_m)$, will be utilized along with the $P(C_m \mid f_k(l))$ for the process of weight calculation.

5.2.2.1. Problem Description

In Figure 5-2(a), the probability distributions of five clusters with respect to the gender of the users are shown. In this example the weights of each gender value can be calculated using (5-3) as follows;

$$w_{sex,\,female} = 0.16^2 + 0.29^2 + 0.12^2 + 0.39^2 + 0.04^2 = 0.278,$$

and similar for the 'Male' feature value.

Note that, the values used to calculate $w_{sex,\,female}$ are read from the Figure 5-2(a). The weight of the 'Female' value, $w_{sex,\,female}$, is larger than the lowest possible value that a weight can obtain, i.e. $1/Q = 1/5 = 0.2$. This means that the importance of the gender value 'Female' is high. On the other hand, for this example, the given $P(C_m)$ in Figure 5-2(b), is almost identical to the $P(C_m \mid f_k(l) = \text{'}Female\text{'})$, which means that, the distribution of the clusters was not affected by adding the dependency on the gender feature values. Moreover, from Figure 5-2(b) it is observed that the distribution of the gender feature values follows almost the same pattern with the clusters' distribution.

Consequently, the gender is not important while clustering the test instances and this fact would not be realized without considering the $P(C_m)$. For this reason, in the proposed two steps weighting method $P(C_m)$ was also taken into account.

If the distributions of the clusters are independent of the feature value of interest, then condition among the feature value no longer provides any valuable information, and

$$P(C_m \mid f_k(l)) \approx P(C_m) \tag{5-10}$$

Therefore, $f_k(l)$ can be categorized as an irrelevant attribute value for the clustering process and $P(C_m \mid f_k(l))$ will follow exactly the same distribution as $P(C_m)$. This also explains the similarity of Figure 5-2(a) and Figure 5-2(b).

5.2.2.2. MDC-BC Algorithm

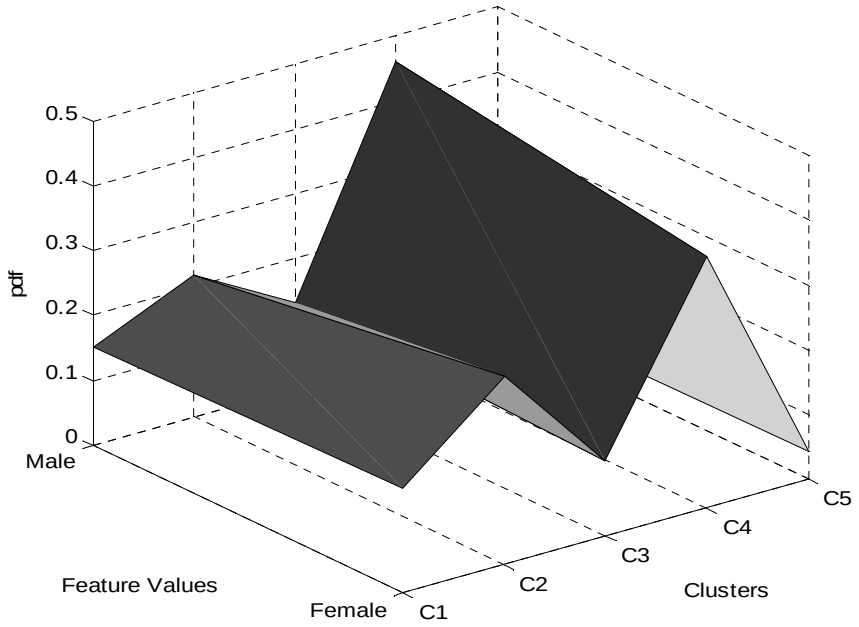
In this method, it aims to achieve accurate feature weight assignment for MDC to obtain better clustering accuracy performance compared to the MDC-CC.

Therefore, Equation (5-3) was modified as follows:

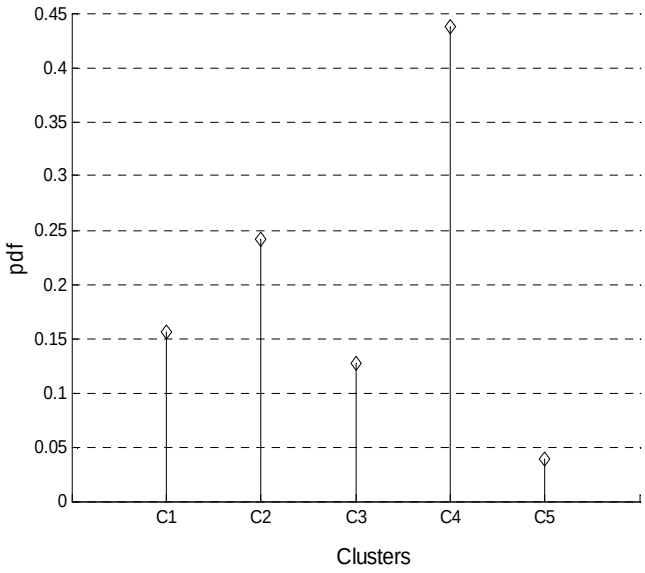
$$\gamma_m = \frac{P(C_m \mid f_k(l))}{P(C_m)}$$

$$w_{k,l} = \sum_{m=1}^{Q} \left(\frac{\gamma_m}{\sum_{n=1}^{Q} \gamma_n} \right)^2 \tag{5-11}$$

In Equation (5-11), the term $\sum_{n=1}^{Q} \gamma_n$ in the $w_{k,l}$ calculation is for the normalization purposes. Figure 5-3 shows the probability distribution using Equation (5-11). The ‘pdf’ of the gender feature values over the clusters are now transformed into a flat distribution. This shows that the ‘Female’ and ‘Male’ feature values appear uniformly across all clusters and so, the gender feature is on its minimum usefulness for the test instance clustering.



(a)



(b)

Figure 5-2 (a) Probability distributions of the gender feature values over the clusters, (b) The probability distribution of the clusters independent of the feature values

The assignment of the feature weights according to the given criterion in Equation (5-11) will be called MDC weight method by Balanced Clustering (MDC-BC).

If the clusters are equi-probable, where equi-probability for clusters is defined as $P(C_m) = 1/Q$ for $m = 1, \cdots, Q$, so that $\gamma_m = Q \;\; P(C_m \mid f_k(l))$ and

$$w_{k,l} = \sum_{m=1}^{Q} \left(\frac{\gamma_m}{\sum_{n=1}^{Q} \gamma_n} \right)^2 = \sum_{m=1}^{Q} \left(\frac{QP(C_m \mid f_k(l))}{Q} \right)^2 = \sum_{m=1}^{Q} (P(C_m \mid f_k(l)))^2 \qquad (5\text{-}12)$$

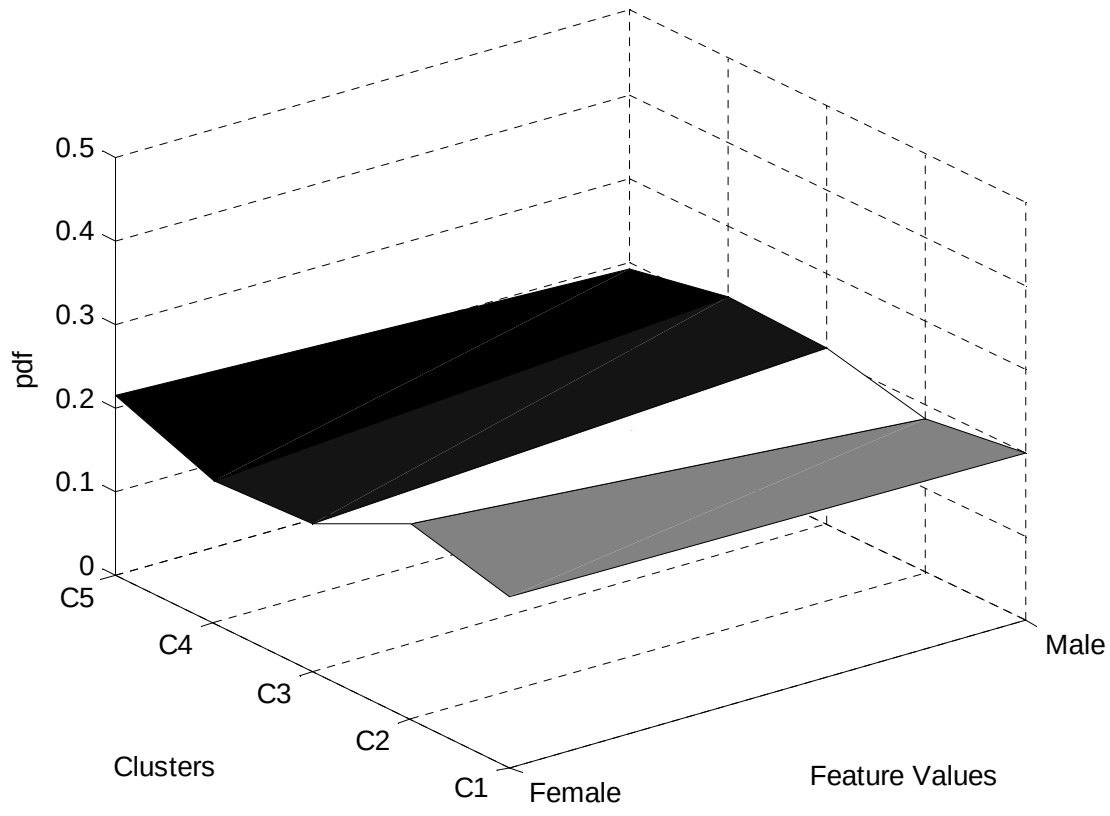


Figure 5-3 Probability distributions of the gender feature values over the clusters with the consideration of cluster distribution

Therefore it can be concluded that the MDC-BC is equal to the MDC-CC method if clusters are equi-probable.

5.2.3. MDC weight method by changing the Lower-limit to Zero (MDC-LZ)

In this method the feature value weights calculated according to (5-11) are minimum of $\frac{1}{Q}$. However, to minimize the effect of irrelevant feature values during the clustering, the minimum value can be decreased to 0. This can be performed by deducting $\frac{1}{Q}$ from the resulted weights. This also enables the important feature values to have better weights relative to the other feature values. According to this, the new boundaries are set to be; $0 \le w_{k,l} \le 1-\frac{1}{Q}$. The modified version of the BC weight algorithm is called Lower-limit to Zero (LZ), in which Equation (5-11) has been modified as follows:

$$w_{k,l} = \left(\sum_{m=1}^{Q} \left(\frac{\gamma_m}{\sum_{n=1}^{Q} \gamma_n} \right)^2 \right) - \frac{1}{Q} \qquad (5\text{-}13)$$

where γ_m is calculated based on Equation (5-11). In Figure 5-4, the algorithm for the instance clustering and distance calculation has been given. The instance clustering function is run for each test instances (line 2). Each test instance is compared with all the train instances (line 3). The distance between each test instance and training instances is considered for the clustering (lines 4 and 5). Therefore, results obtained from the distance function define which cluster the test instance belongs too. In distance function the calculation is

Input: $X_i = \{x_i(1), x_i(2), ..., x_i(A)\}$ // test instance
$Y_j = \{y_j(1), y_j(2), ..., y_j(A)\}$ // train instance
$w_{k,l}$ // weight matrix
Output: Updated train dataset
Algorithm:
{*Function I: Instance clustering*}
1. q // threshold
2. *for* $i = 1$ *to* M *do*
3. *for* $j = 1$ *to* N *do*
4. *if* $(dist(X_i, Y_j) < q)$
5. Cluster X_i using Y_j's cluster label
6. *end if*
7. *end for*
8. *end for*

{*Function II: Distance Calculation*}
9. $dist = 0$
10. *for* $k = 1$ *to* A *do*
11. $dist = dist + w_{k,l} g(x_i(k), y_j(k))$
12. *end for*
13. *return dist*

Figure 5-4 Algorithm of the instance clustering andand distance calculation functions of the MDC

performed feature by feature (line 10). The weights of feature values, the outcome of the function $g(x_i(k), y_j(k))$ and the 'dist' value are used for the distance computation (line11). Note that the $w_{k,l}$, weight matrix, differs based on the feature weight method. Therefore, the $w_{k,l}$ is calculated before the instance clustering making use of the training dataset by Equation (5-3), (5-11) or (5-13) and feed into the algorithm.

5.3. Implementation and Evaluation of the MDC

A set of computer simulations was carried out to validate the performance of our proposed MDC methods for user profiling. Subsection 5.3.1. describes the datasets used for the simulations while Subsection 5.3.2. presents the results gathered from these simulations.

5.3.1. Dataset

For the simulations the dataset used was provided in [80], named 'Adult Dataset'. This dataset was created by Barry Becker via extracting information from the 1994 census dataset and denoted to UCI (University of California, Irvine) Machine Learning Repository [80] by Ronny Kohavi and Barry Becker for data mining applications. In this dataset the demographic information of 32500 users is listed, which has been adopted as a draft to create a complete dataset of user profiles for the simulations. A total of 10 features of the demographic information of the users were selected from this dataset which are: Nationality, Sex, Age, Marital Status, Origin, Employment, Profession, Education, Relatives and Annual Income. Four more features, highly correlated to the user clusters, were created reflecting the interest profile and preference profile of the users, which were: Sport, Book, Leisure-preference and Music interests. Therefore, each user represented with three sets of profile information, namely demographic, interest and preference data. The training and test datasets have been selected from the complete user profile dataset. Note that, unlike the traditional 'k fold cross-validation', here dissimilar training and test datasets have been used that include information of different users. For the simulations each of the three algorithms trained on the same training set and tested on the same test set.

The simulation parameters were set to be $Q=14$, $C=5$, $N=10000$, $M=1000$.

5.3.2. Simulation Results

The first simulations are carried out using IBL. In Figure 5-5 it is shown how the error rate changes as the number of training instances changes, if IBL is used to classify the user profiles. Here, the error rate is the percentage of wrongly classified test instances over M, $\mathrm{E}_{IBL}=\frac{number\_of\_incorrectly\_clustered\_inst}{M}*100$, where E_{IBL} is the error rate for the IBL. In Figure 5-5 three plots have been presented and all tend to decrease as N increases.

The first plot, shows the error percentage when the IBL is run only over the demographic data of the users. The second plot, in the middle, shows the second best performance and represents the simulation that carried out with the interest profile of the users. The better performance of the interest profile is due to the selection of the interests to be highly correlated to the clusters. The third plot is the performance of IBL if all profiles (interest, preference and demographic) are used to classify the users. This plot flattens at an error of approximately 35%. We note that the level of the error reflects the relationship between the features and the classes. It is expected that the use of more relevant features or increasing the Q will further lower the error level.

The simulation results of weighting methods “CC”, “BC” and “LZ” introduced in MDC algorithm are shown in Figure 5-6. The simulations were conducted to monitor the error rate versus the increasing number of training instances. It can be seen from Figure 5-5 that the minimum error percentage gained is 35% when IBL is used. In Figure 5-6 the $\mathrm{E}_{IBL}=35\%$ was chosen to be the reference

point representing 0%. The aim was to determine the effect of the weighted MDC's on the error rate of the IBL.

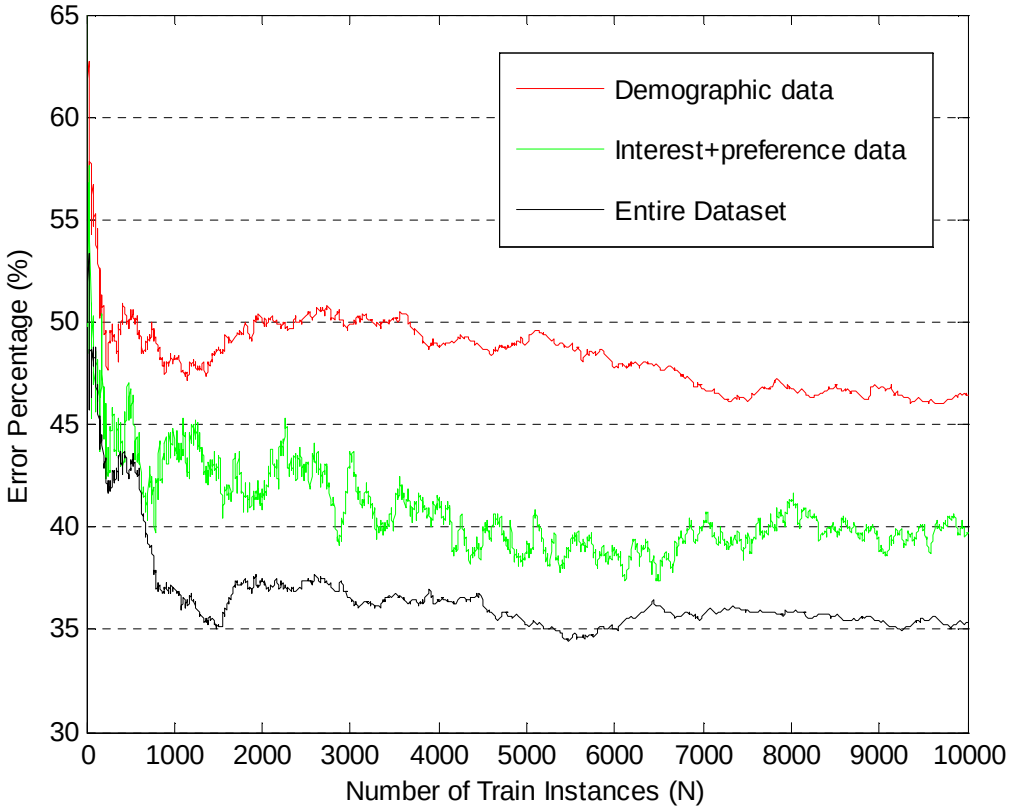


Figure 5-5 The change in the error percentage as the number of training instances increases

In Figure 5-6 the equation $R_{error} = (1 - \frac{E_{MDC}}{E_{IBL}}).100$ has been used to find the relative error percentage of the weighted versions of the MDC. The value of E_{MDC} was calculated similar to E_{IBL}. In Figure 5-6 each plot shows the level of improvement in the error rate relative to the minimum error rate that is obtained when IBL is used for the user profiling.

It can be seen from Figure 5-6 that MDC-CC's relative error percentage saturates to approximately 17% as the training instances increase over 10000.

Therefore, MDC-CC decreases the error percentage by 17% compared to the IBL. Also one can observe from the plot that for the BC method, the relative error percentage further improves by approximately 20%.

The dissimilarity between the MDC-CC and MDC-BC increases significantly approximately after the 7500th training instance. This means that after the 7500th training instance each of the two MDC versions saturate into a level where increasing number of the training instances does not contribute to any further changes to the error rate. The relative error percentage distribution of the MDC with the LZ weight method is shown on the top plot in Figure 5-6, and

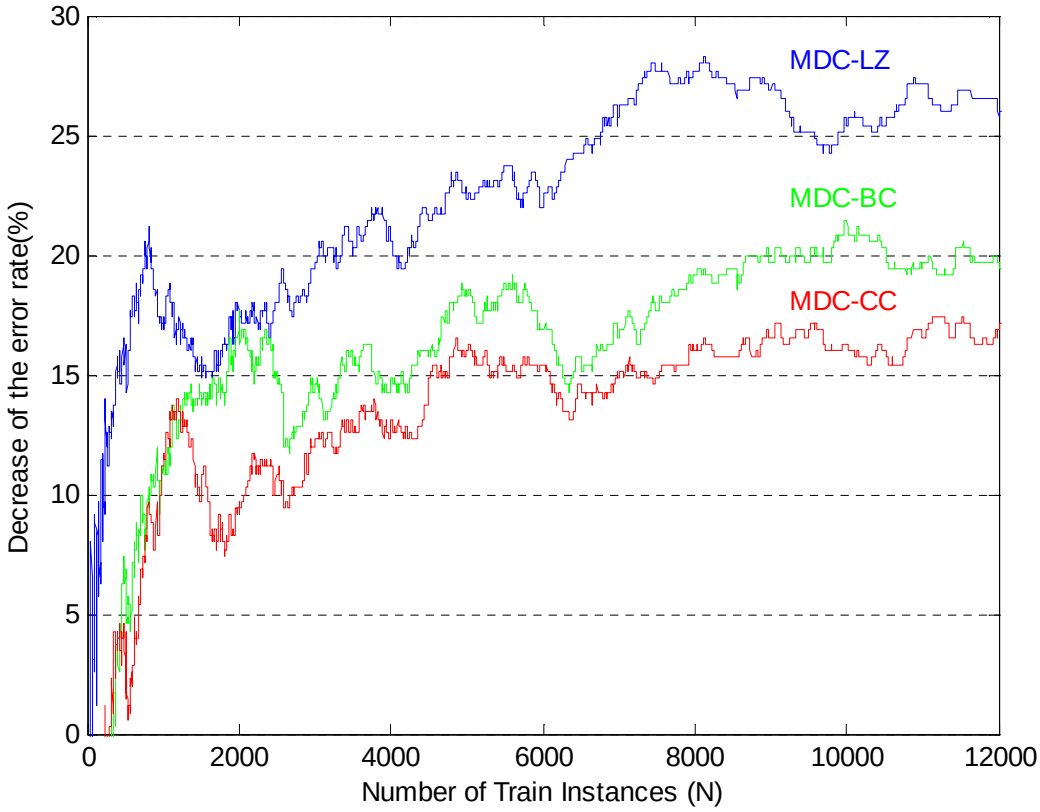


Figure 5-6 The improvement to lower the error rate by introducing weighting MDC for the user profiling

represents the lowest relative error rate that has been achieved compared to MDC-CC and MDC-BC with an improvement of approximately 27%. The plots

show that MDC-CC and MDC-BC results are closer than MDC-BC and MDC-LZ. This result was expected as both MDC-CC and MDC-BC have a similar feature value weighting scheme.

Comparing Figure 5-5 and Figure 5-6 shows that in general the weighted versions of the MDC give better classification accuracy results than IBL. This is because: when compared to the IBL in the weighted MDCs, relevant features have a higher impact on the clustering while irrelevant features have lesser impact. Note that the weights of the feature values define the level of relevance of the features for the clustering. Here the weights close to $\frac{1}{Q}$ are defined as irrelevant while those that are close to 1 defined as relevant feature values, according to (5-3).

Furthermore, test datasets of different sizes were also simulated with the proposed methods, for $M=1000$, $M=1500$ and $M=2000$ where M represents number of test instances (see Subsection 2.1.2.3.). The simulation results for IBL and MDC-LZ algorithm are shown in Figure 5-7 and 5-8 respectively. All values of M produce similar results saturating to approximately 35% for IBL in Figure 5-7 and to 25% for MDC-LZ in Figure 5-8.

In order to make use of PCF for user profiling the following two methods were proposed:

Method 1: In order to make correct clustering more likely to occur, the probability values of the weight calculation can be inverted as follows:

$$w_{k,l}(C_m) = 1 - P(C_m \mid f_k(l)) \tag{5-14}$$

Method 2: A second way of avoiding PCF to generate incorrect clustering is by modifying the minimum distance criterion of IBL. The cluster of the training instance with the maximum distance, using the function in (3-6), can be taken as the correct cluster value as

$$\arg\max_{j} dist(X_i, Y_j) \text{ for } i = 1,2,3,...,M \ . \tag{5-15}$$

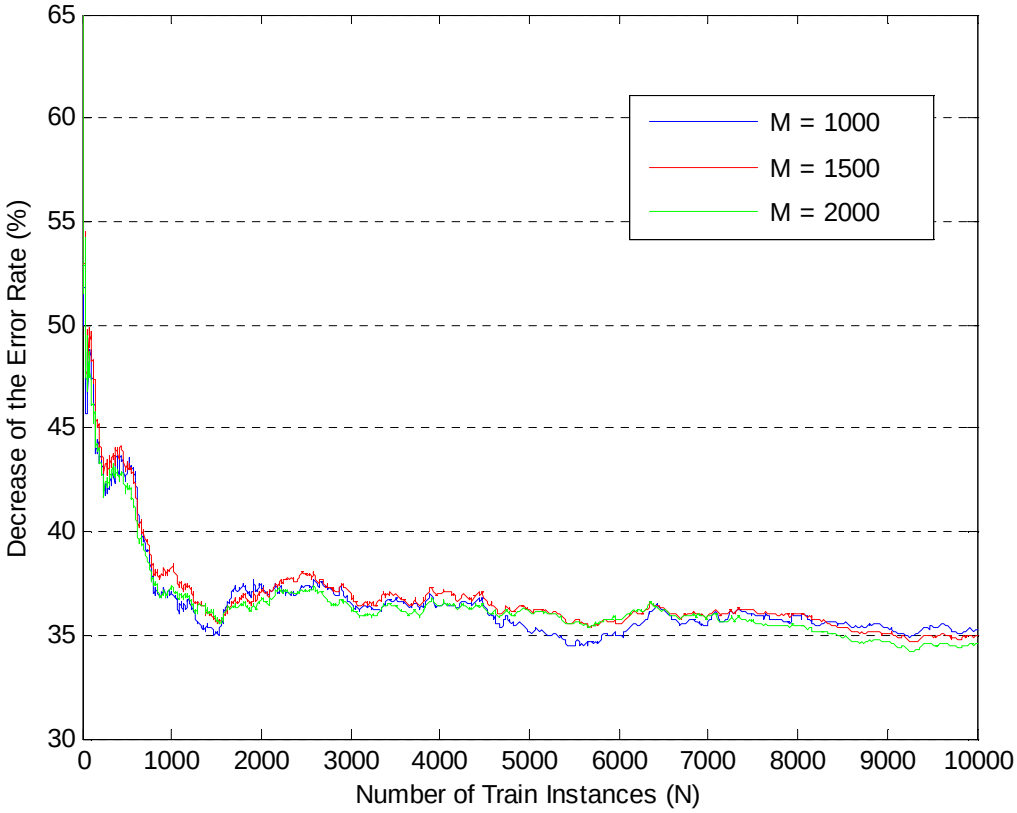


Figure 5-7 The performance of the IBL algorithm over the test datasets of three different sizes

Although the two proposed methods benefit the PCF's lacking clustering performance, it has been realized through the simulations that the performance is still not very promising after the modifications.

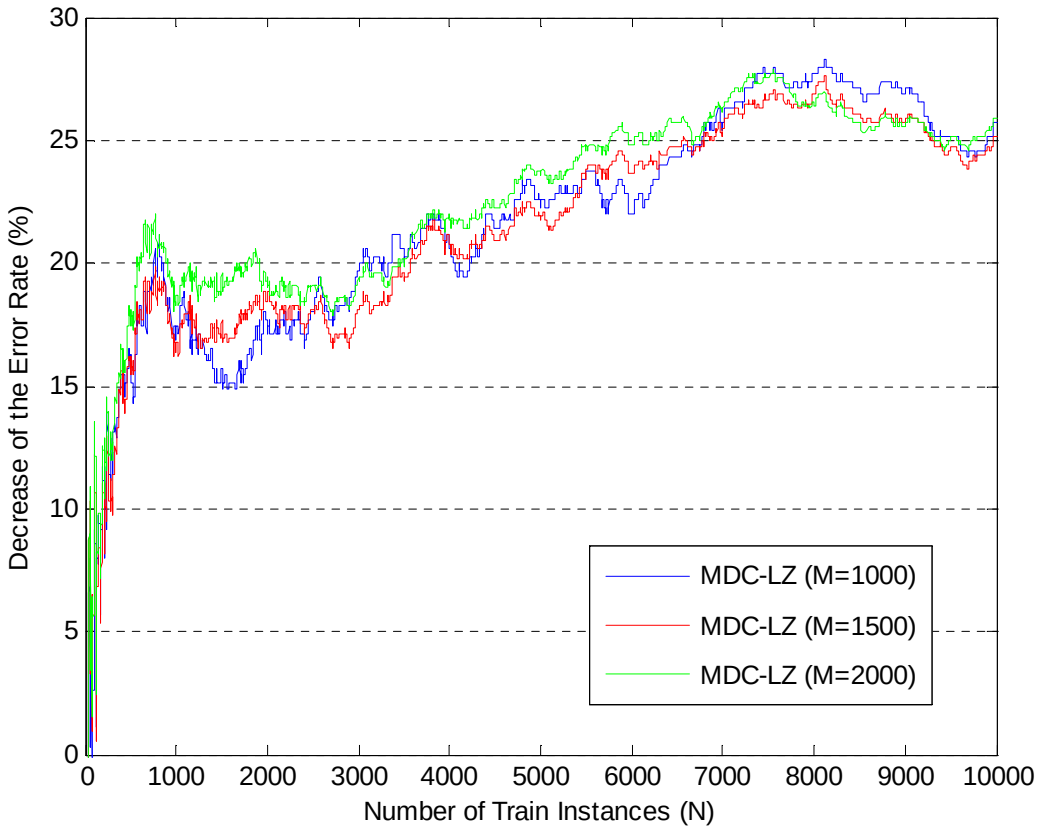


Figure 5-8 The performance of the MDC-LZ algorithm over the test datasets of three different sizes

This is because the new equations (5-14) and (5-15) are only to invert the incorrect clustering performances and utilizing these modifications for the first method degrades the structure of PCF while in the second method the structure of IBL was degraded. The simulation results are depicted in Figure 5-9. Here the top subplot shows the error performance of method 2, where the error rate is in the margins of 43%. The bottom plot shows the decrease on the error rate of IBL when method 1 is used. By comparing Figure 5-9 with Figure 5-5 it can be seen that the decrease is even lower than 10%.

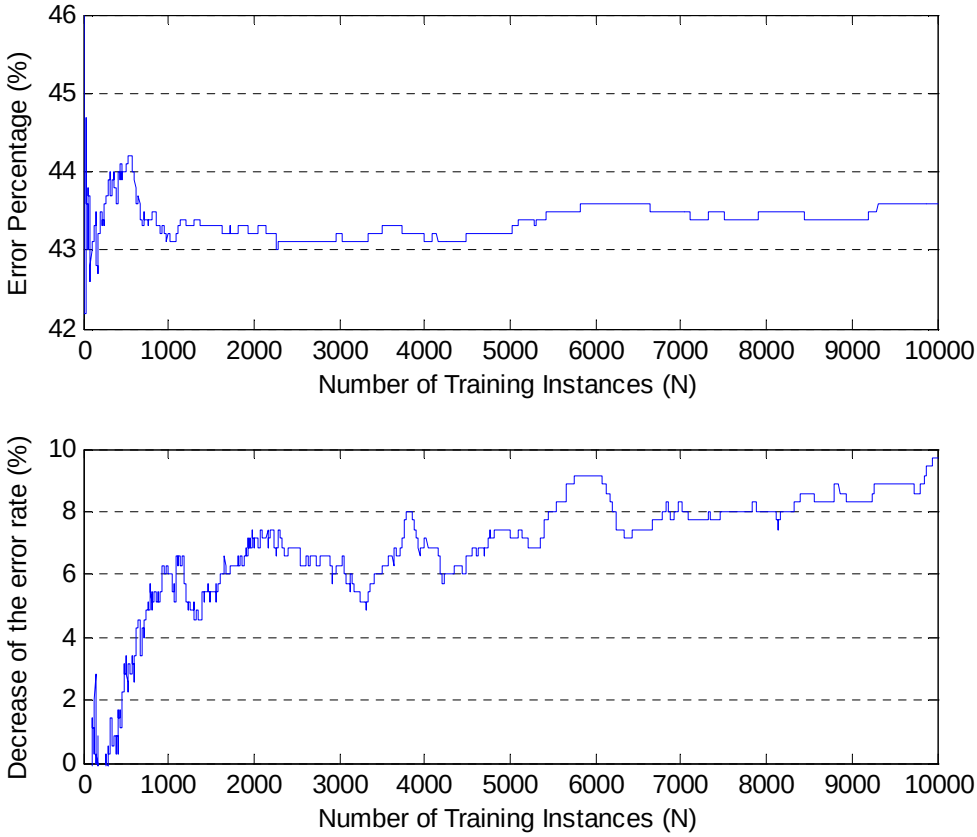


Figure 5-9 PCF's error performance with method 1 and method 2

5.3.2.1. Comparison with the Existing IBL Algorithms

In the literature various weighted and non-weighted IBL algorithms have been proposed. IBK is one of the well known IBL algorithm where, different from IBL, K closest instances are retrieved and the label of the majority class among these instances is assigned as the class label for the test instance [68][106]. The following paragraphs are associated with IBK:

- If the class attribute is symbolic then the class label of the test instance X_i is the same as the class label of the highest vote among the K nearest neighbours. For a scenario, where $K=3$, if the three nearest

neighbours Y_1, Y_2 and Y_3 belong to the classes, C_1 , C_1 and C_2 respectively, then C_1 is assigned as a class label for X_i, $X_i \in C_1$, since C_1 is the predominant class label among nearest neighbours.

- If class attribute is numeric then the class label of the test instance will be the mean of the nearest neighbours. Following the above assumptions, label of X_i, $L(X_i)$, is calculated as;

$$L(X_i) = \frac{\sum_{m=1}^{K} L(Y_m)}{K} \tag{5-16}$$

where $L(Y_m)$ represents the class label of Y_m. Two of the well known weighted versions of the IBK algorithm are “distance weighted IBK with (1/d)” (dw-IBK (1/d)) and “distance weighted IBK with 1-d” (dw-IBK (1-d)). (1/d) represents the weight obtained from the inverse of the distance (1/d) whereas (1-d) means that the weight is obtained by subtracting the distance from a constant (i.e. 1) [106][107]. Here, if the result of the subtraction is greater than zero than the weight is the result; otherwise the weight is zero [108]. In case of numeric class attribute, Equation (5-16) is modified as follows;

$$L(X_i) = \frac{\sum_{m=1}^{K} w_m L(Y_m)}{\sum_{m=1}^{K} w_m} \tag{5-17}$$

where,

for dw-IBK (1/d) w_m is,

$$w_m = \frac{1}{dist(X_i, Y_j)^2} \quad (5\text{-}18)$$

for dw-IBK (1-d), w_m is,

$$w_m = 1 - dist(X_i, Y_j)^2 \quad (5\text{-}19)$$

Locally Weighted Learning (LWL) algorithm is a weighted IBL that assigns weights to instances using IBL and uses these locally weighted training instances for classification [109]. While the IBK performs local approximation for each test instance X_i, LWL performs an explicit approximation of $L(X_i)$ for region surrounding X_i by fitting linear function and quadratic to K nearest neighbours.

KStar (or K*) instance based learner was proposed by Clear *et al.* [89] and aims to provide a consistent approach to handle symbolic attributes, real valued attributes and missing attributes [90]. K* is based on entropy distance measure where the distance between two instances is defined as the complexity of transforming one instance into another [89] [107]. This complexity calculation is done in two steps. First the finite set of transformations that map instances to instances is defined. A 'program' which transforms one instance X_i to another instance Y_j is a finite sequence of transformations starting at X_i and terminating at Y_j. Kolmogorov is one of the well known entropy distance where the distance between two instances is the shortest string connecting them. Hence, this approach is focused on the shortest transformation out of many possible transformations. Here, the resulted distance measure is very sensitive

to small changes. For this problem K* is defined as the distance by summing all possible transformations between two instances.

Let assume,

I as set of instances,

T a finite set of transformations on I ,

t_i being an element in T , maps instances within I , i.e. $t : I \rightarrow I$,

Based on the above assumptions, K* function can be defined as:

$$K^*(X_i / Y_j) = -\log_2 P^*(Y_j / X_i) \tag{5-20}$$

where,

$\bar{t}(X_i) = t_n(t_{n-1}(..t_1(X_i)...))$ and $\bar{t} = t_1,...,t_n$ [89]. Here the probability function P^* is defined as the sum of the probability of all paths from instance X_i to instance Y_j [89].

$$P^*(Y_j / X_i) = \sum_{\bar{t} \in P.\bar{t}(X_i)=Y_j} p(\bar{t}) \tag{5-21}$$

Table 5-1 compares the MDC-LZ with the above mentioned algorithms in terms of 1) the distance metric, 2) number of neighbours involved in classification, 3) weighting function, 4) how the label prediction is done and 5) error rate. It can be seen from the table that, except for Kstar, other algorithms use Euclidian function to calculate the distance between instances. It can also be observed that IBL, LWL and MDC-LZ are similar as all of these three algorithms consider the closest neighbour to predict the label for the new instance.

To compare the performance of MDC-LZ against existing weighted and non weighted IBL algorithms, a set of computer simulations were carried out. WEKA [107] was used as the simulation platform and the simulation parameters were

set as default by WEKA except the K value being taken as 2. The previously defined training and test user profile datasets (see Subsection 5.3.1.) have been used for all the algorithms.

Table 5-1 shows the error performance results of MDC-LZ, IBL, IBK, dw-IBK (1/d), dw-IBK (1-d), KStar and LWL. MDC-LZ has achieved the lowest error rate. Second best result was obtained with KStar. Table 5-2 also shows that MDC-LZ performed better than IBK in terms of error rate. Hence, for user profiling, using K nearest training instances for clustering is not as effective as weighting. LWL achieved the worst performance among other classifiers.

IBL and its variants have computational complexities in the order of $N \times A$. As can be seen from the pseudo-code given in Figure 5-4, the MDC methods proposed in this chapter are no different. As long as the aim is to compare every feature of every training instance, the order of the computational complexity will always be $O(NA)$, where $O(\cdot)$ represents the order.

Of course having complexities in the same order does not mean that they all constitute the same number of operations. In IBL, for every dissimilar feature, the distance is increased by the value of the distance function given in (3-4) and (3-5). On the other hand, MDC methods need one multiplication per dissimilar feature formulated in Equation (5-1), before the distance is calculated. Therefore, MDC requires extra operations to perform (5-1) in addition to the computational cost of IBL. Based on this assumption the computational complexity of MDC, D_{MDC}, is defined as:

Table 5-1 Comparison of weighted and non-weighted IBL algorithms

Algorithms	Distance Metric	Number of neighbours to look at	Weighting Function	How to fit the instance	Error Rate (%)
Existing Algorithms					
IBL [68] [69]	Euclidian	one	Unused	Predict the same label as the closest instance	35%
IBK [68] [106]	Euclidian	K	Unused	Predict the same label as the predominant K closest instances	35.8%
IBK-dw(1-d) [106] [107]	Euclidian	K	1-distance	Predict the same label as the predominant K closest instances	34.7%
IBK-dw(1/d) [106] [107]	Euclidian	K	1/distance	Predict the same label as the predominant K closest instances	34.7%
LWL [109]	Euclidian	One	Linear Regression	Predict the same label as the closest instance	42%
Kstar [89]	Entropy Distance - Kolmagorov	K	Unused	Predict the same label as the predominant K closest instances	30.9%
Proposed Algorithms					
MDC-LZ	Euclidian	One	Lower-to-Zero (LZ)	Predict the same label as the closest instance	25.55%
MDC-CC	Euclidian	One	Cross Clustering (CC)	Predict the same label as the closest instance	29.05%
MDC-BC	Euclidian	One	Balanced Clustering (BC)	Predict the same label as the closest instance	28%

$$D_{MDC} = D_{IBL} + \sum_{j=1}^{N} Z_j << D_{IBL} + N \times A \quad (5\text{-}22)$$

where D_{IBL} is the computational complexity of the IBL and Z_j is the number of features that the j th training data has, which are different from those of the test instance. Equation (5-22) shows that D_{MDC} needs $\sum_{j=1}^{N} Z_j$ operations (that includes reading from the weight matrix and a multiplication if the dissimilar feature is numeric) more than IBL which is always less than $N \times A$ operations. Therefore the computational complexity of MDC can still be represented as $O(NA)$.

Apart from the clustering stage, the calculation of the weights used to calculate the distance, also requires extra computations. However, this is performed only once when the system is set up and updated regularly, and therefore any complexity arose from this stage can be ignored.

5.4. Case Study

Today, mobile device users receive a variety of services and information delivered to their mobile devices. Many of these are irrelevant, far from the user's satisfaction level and may likely be regarded as spam messages by the user. This results in the users to look for the relevant services by themselves which would be time consuming and may cause dissatisfied customers.

In this section we present a scenario which demonstrates the use of a multi-dimensional clustering algorithm for the user profiling to improve personalized service provisioning in mobile environments.

5.4.1. Proposed Scenario

In this scenario we focus on a mobile advertising service. Here we introduce a personalized mobile advertising service called Discounts, Promotions and Deals (DPD). DPD advertising service provides discount, promotion and deal advertisements to the user according to the user's profile. Furthermore, for this case study, DPD is concerned with the food industry, and a restaurant service called MyRestaurants, has been chosen. The following user is assumed for this scenario.

Ren is a 30 years old Londoner. She is working as a property adviser in a company located in central London. She has got an iPhone™ and a BlackBerry™ smartphones which have been provided by the company. She uses her BlackBerry™ for work related duties while her mobile phone is a part of her personal life.

Ren decided to subscribe for the personalized mobile advertising service, MyRestaurants. Recently the following advertisements have been announced:

- EFES-2™, Turkish restaurant in central London, has meal deals where order of a 3-course meal for two comes with a free bottle of wine
- Gourmet Burger Kitchen™, Soho branch in central London, has 2 burgers for £10.
- Bella Italia™ restaurant, Covent Garden branch in central London, has a 30% discount when 3-course meal is ordered.

Through her mobile device, each of the advertisement is presented with the link where a user can follow for more information.

Ren prefers to receive the advertisement everyday and likes to check it out the ads in the morning time. Subsequently, on Monday morning, around 9am on her way to work, Ren signs into the MyRestaurants service through her iPhone™. She receives the advertisement listed above. She is pleased with the EFES-2™ meal deal offer as this restaurant is very close to her work place and she has previously thought about trying out its food. Ren follows the provided link to book a table through the restaurant's mobile-web.

5.4.2. System Overview

The following four subsections explain the architecture of the proposed system, user learning, user profiling and restaurant recommendation for this case study. Figure 5-10 shows the flowchart of the user learning and profiling. User learning process starts whenever the user signs into the MyRestaurants. Here, the system monitors user's feedback towards the given recommendations until user signs out from the system (i.e. session terminates). Following this, the new information from the learning process is used for the user profiling. In this process, a clustering algorithm (MDC-LZ) will update the user's profile information in the user profile dataset with using the information from user learning process. The following subsections (Subsection 5.4.2.2. and Subsection 5.4.2.3.) give more detailed information on both aforementioned processes.

5.4.2.1. Architectural Model of the Proposed System

This subsection provides detailed information about the proposed architecture for personalized mobile service provisioning for this case study. The architecture is shown in Figure 5-11 and it includes six main parts. These are

the user profiling centre, personalization and recommendation centre, privacy manager, context manager, service provider and device manager.

The user profiling centre consists of two processes. The first process, user learning, starts when user signs into the system and ends when user signs out of the system. Here, new information about the user is learned by monitoring the user-system interaction via mobile device. The outcome of the user learning is used for the user profiling process by the MDC-LZ algorithm. User profiles are the outcome of this second process and they are stored into the user profiles DB. More detailed information about the above mentioned processes is given in Subsections 5.4.2.2. and 5.4.2.3.

In personalization and recommendation centre there are three inputs to the service personalization process. These are coming from the user profiles DB, service retrieval and context management. Service personalization process uses these three inputs to personalize and recommend the location based mobile services to the users. Here, service retrieval fetches the service from the service provider where all the service information is kept. Service provider decides which services to push to the service retrieval based on the information coming from the privacy manager. Subsection 5.4.2.4. provides detailed information about the privacy manager and personalized recommendation.

Each part of the proposed architecture is significant for the successful location based mobile service personalization. Moreover, deployment of the whole architecture is a large scale project. Hence, it is worth pointing out that the investigation of the user privacy issues, device management, personalization and context management is considered out of the scope of this research.

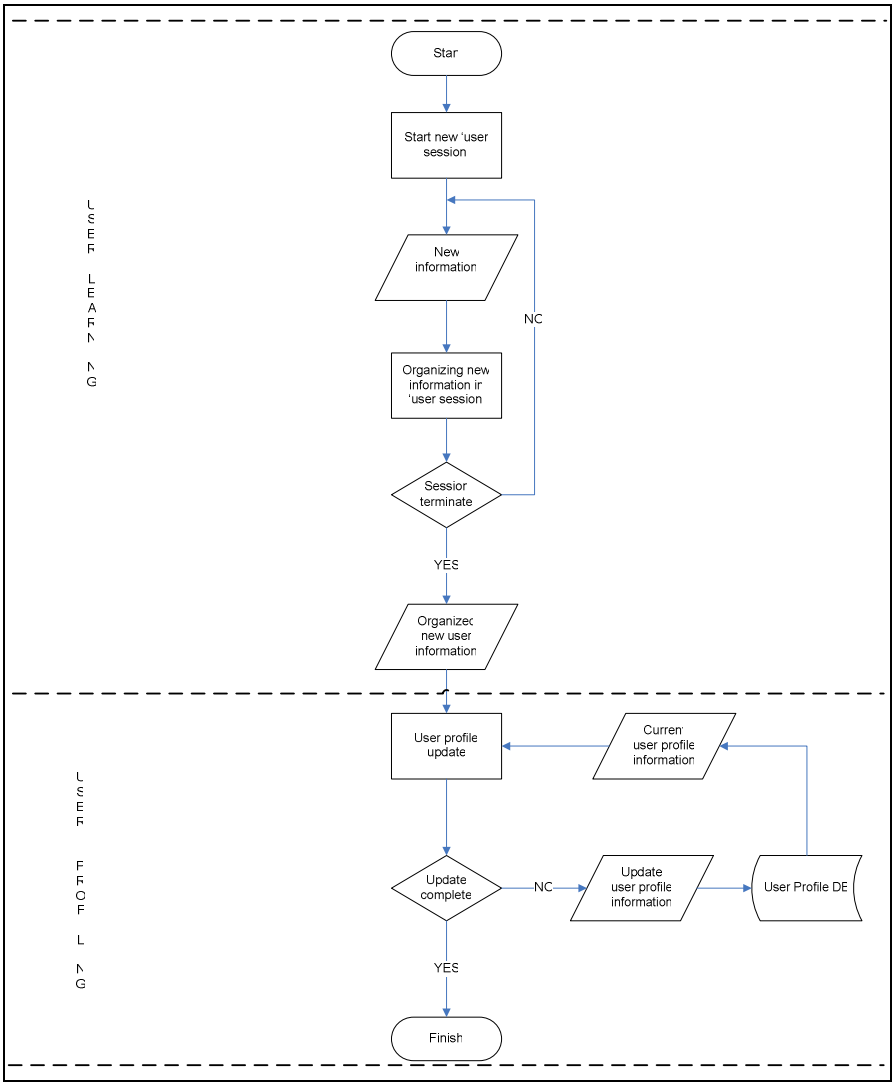


Figure 5-10 Flowchart of the user learning and user profiling

5.4.2.2. User Learning

For this case study we assume that the information given by the user during the subscription is to be used for the initialization of the user's profile. Note that this corresponds to the directly/explicitly information gathering that we discussed in Chapter 2. The user's response (user feedback) to the provided services will then be used to update the user's profile implicitly. It is worth pointing out that the location preference of the user will be kept in the user profile. Each user will have an identification (i.e. user-id and password) for the purpose of authentication for the service.

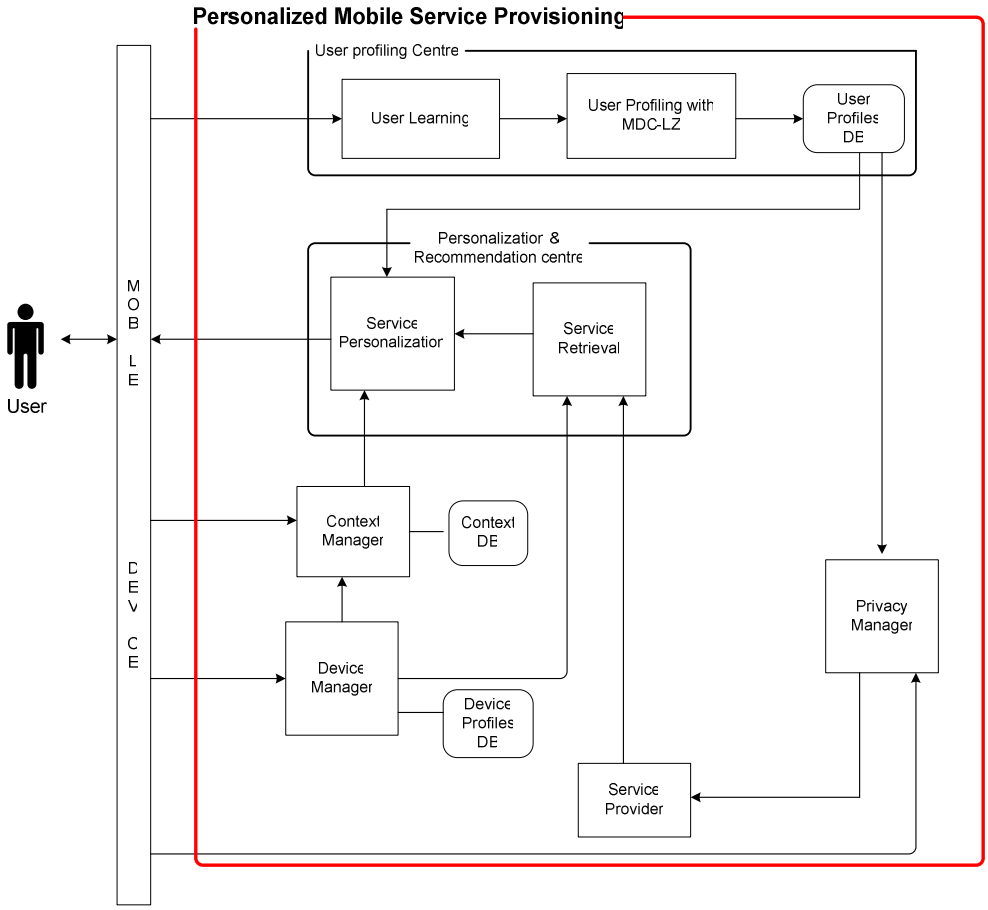


Figure 5-11 Architecture for personalized mobile service provisioning

Here, the system will automatically assign Ren a user-id and a password when she subscribes for the service. An initial password can be changed by the user following first sign in.

After subscription and registration, the system continuously monitors Ren's feedback and behaviour towards the provided services to learn more about her (i.e. what services she likes, when and where). For example, monitoring Ren shows that she prefers to receive the advertisements every morning while travelling to work.

5.4.2.3. User Profiling

For this case study the MDC-LZ is used for the user profiling. Here, MDC-LZ will assign different weights to the user profile attributes to increase the impact of relevant attributes in clustering so as to define the user's service preferences more precisely. The data flow in and out of the MDC-LZ algorithm is shown in Figure 5-12. It can be seen from this figure that there are two inputs to the MDC-LZ, test data and training data. The new user information is referred to as test-data while training-data is the existing user information.

The output from the MDC-LZ is the clustered test-data, which becomes a training-data following processing by the MDC-LZ. In MDC-LZ each feature has a weight and the weight matrix, constructed from the feature weights, is used for the distance calculation and instance (user) clustering. In MDC-LZ a weight is assigned to each feature via a LZ feature weighting method.

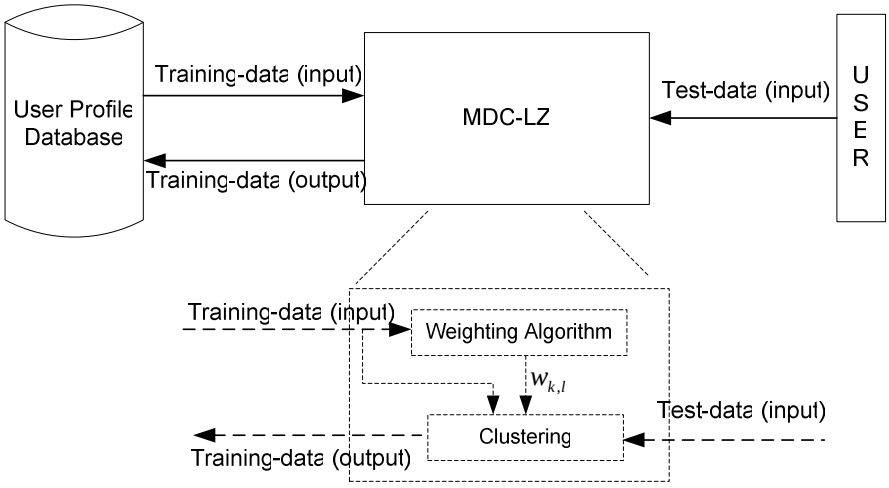


Figure 5-12 MDC-LZ Data flow

Which user receives which advertisements is decided by making use of the user's profile information and the cluster that the user belongs to. In this way, the same advertisements can be sent to the users that share the same cluster

and these users receive the advertisements that most of the users in the same cluster showed a liking for.

User's location preference and user's current location are two important parameters for providing the right location based advertisements. For example, when it comes to the location based advertisements, Ren prefers the ones that are close to her work place so her location preference is 'work'. However, she is a property adviser and she needs to travel to different UK cities very often. Hence, when Ren is away, she will receive location based advertisements based on her user profile information and current location rather than her preferred location. The current location can be extracted from the GPS (Global Positioning System) information of the user's mobile device.

5.4.2.4. Restaurant Recommendation

Many works in the literature show that the mobile recommendation becomes very popular due to the growing diversity, availability and use of mobile information services [110]. For personalized mobile services, various architectures have been proposed [111]-[113]. Referring to Figure 2-3, personalized restaurant recommendations are the outcome of the personalization process. In this case study personalization process uses user profile information to personalize (filter) the restaurants to be recommended to the user. In Figure 5-11 detailed information of this process is shown. From this figure it can be seen that there are three inputs to the 'service personalization'. These are user profile, service to be personalized and current context information. Context information (i.e. location) and device capabilities are obtained from the mobile device. These are considered to be important for

accurate user interface adaptation and personalization. Here, a privacy manager uses user's sign in information and user profile information to decide who can use the user profile information for what purpose, who the user is and if they have the right to use the provided service. It is worth to point out that, like each user, each restaurant has to subscribe to MyRestaurant to be recommended to the users. This means that service provider acts like a bridge between users and restaurants.

Here, a user profile dataset, which has been defined in Subsection 5.3.1., is used. Figure 5-13 is an example of some of the demographic, interest and preference information of a user in user profile with the following order; Age, Annual Income, Sex, Sport Interest, Music Interest, Book Interest, Leisure, Marital Status, Employment, Education and Profession.

The MDC-LZ uses this given data to predict the user's cuisine preferences. Here, user's cuisine preference is represented with its probabilistic distribution function which enables the user to receive recommendations from different types of restaurants. For instance, user's cuisine preference can be 40%Turkish, 30% British, 20%Italian and 10%American.

53	<10	Male	Football	Rock	Political	Nature-Trekking	Married-civ-spouse	Private	11th	Handlers-cleaners

Figure 5-13 Example of user profile information

These probabilities can change based on the users feedback to the given recommendations. In this study the user's clicks on a given recommendation is considered as a positive feedback. Here, the system counts each click on

recommended restaurants and utilize this information to update the user's cuisine preferences. Therefore, user's current information and new information are incorporated together as shown in Figure 5-12 to update the user profile information.

As mentioned previously, the user's location preference information (home, work or elsewhere) is also kept in the user profile and used for the location based restaurant recommendations. Gasson *et al.* [114] showed what kind of personal information can be obtained by monitoring a user's mobile device while in [110] it has been shown how the GPS data can be converted into text format. This method makes it possible to compare restaurants' location and user's location preference (or user's current location in case of elsewhere) to provide accurate recommendations.

Similar to the user profile dataset, restaurant information is kept in the restaurants dataset. Figure 5-14 shows an example for the restaurant profile information. In this separate dataset each restaurant is represented with the following attributes: Name, Cuisine Type, Price, Deal Description and Location. Here, each of these are used to classify restaurants based on their cuisine types using IBL (see Chapter 3).

EFES-2	Turkish	<30	order of 3-course meal for two comes with a free bottle of wine	175-177 Great Portland Street London W1W 5PJ

Figure 5- 14 Example of restaurant profile information

5.4.3. Implementation of the proposed scenario

This section implements the proposed scenario and shows the usage of a DPD-Restaurant application, named MyRestaurants, from the user's point of view. The scenario is implemented as a Java Mobile Application (Java ME) on NetBeans IDE 7.1. Note that for this scenario we assume that user Ren is already subscribed for the service.

Following her subscription, Ren started using the service. To check her restaurant recommendations she needs to signs into the system using her user-id and password (see Figure 5-15). Here, prompt information is compared with the information in the user's profile for authentication.

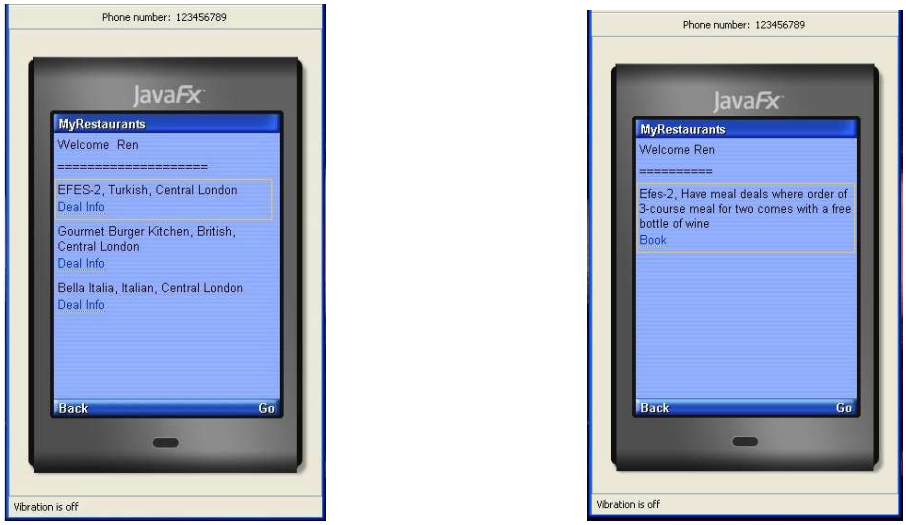
Ren's successful sign-in redirects her to the MyRestaurants main page. This main page displays two options: 'My Account' and 'My Deals'. First option, 'My Account', redirects her to a new page where she can change her password, location preference and user-name. The user-name is different from the user-id and it is used for display purposes. In this scenario she prefers her user name to be 'Ren'.

'My Deals', on the other hand, redirects her to a new page. This new page includes daily restaurant recommendations (see Figure 5-16). Each recommendation has a link which provides more information about the deal and the restaurant (see Figure 5-16). Here, if she wants, she can follow another provided link to make a booking.

The implementation of the above scenario aimed to show how proposed user profiling algorithm can be used for the restaurant recommendation via mobile

devices. Furthermore, it is worth to mention that because MDC: is designed by considering the multidimensionality of the user profile, and is implemented on Java platform, any third party service provider can use this algorithm to provide personalized services/recommendations with maximum possible user profile accuracy.

Figure 5-15 Ren enters her user-id and password to sign-in



(a) **(b)**

Figure 5-16 (a) Ren's daily restaurant deals, (b) Detailed deal information

5.5. Summary

In order to lower the effect of irrelevant features and increase the effect of relevant features in the clustering process a clustering algorithm named Multi-dimensional Clustering (MDC) has been proposed by the author for user profiling. MDC is a modified version of the IBL, and it assigns weights to the feature values and does clustering of the users based on the weighted distances. Three weighting methods were proposed for the MDC that are named Cross Clustering (CC), Balanced Clustering (BC) and Lower-limit to Zero (LZ). A set of computer simulations was carried out to validate the performance of the proposed methods for user profiling. The evaluation of the results was made based on the clustering accuracy and error percentage. All the four algorithms, MDC-CC, MDC-BC, MDC-LZ and IBL, were trained and tested on the same datasets. The results presented in Figure 5-6 show that each of the three MDC versions improves the error rate of the IBL. In this chapter the use of Per Category Feature (PCF) weighting for the IBL was also investigated and evaluated. Obtained simulation results were indicated that the PCF is less effective when it is used for the purpose of multi dimensional clustering for user profiling. Additional simulations were carried out with weighted and non-weighted IBL algorithms in the literature that are IBK, dw-IBK (1/d), dw-IBK(1-d), KStar and LWL. The results in Table 5-1 showed that the proposed MDC-LZ achieved the lowest error rate among other algorithms.

The last section of this chapter presents a case study example. In this case study a real life scenario is implemented as a Java Mobile Application (Java ME) on NetBeans IDE 7.1. The aim of this application was to show how the

multi-dimensional clustering algorithm can be used for the user profiling to improve personalized service provisioning in mobile environments.

Chapter 6

Evaluation, Conclusions and Future Works

In this chapter, evaluation, conclusions and future works for this thesis are given. The evaluation section summarises the research work carried out by pointing the problems and solutions. Following this, the main conclusions from each chapter are presented in Section 6.2. Finally, possible future works are given in Section 6.3.

6.1. Evaluation

Today a large number of services are available for customers using the online-facilities on the web which escalates the competitiveness within the market. In this competitive environment it is a major challenge for the service providers to survive. Personalization of services is an opportunity to help to improve quality of service. Hence, many application areas intend to have optimum user satisfaction via personalization. The success of these applications rely on how well the service provider knows the user requirements and how well this can be reflected on the services. The description of the user interest, preferences,

characteristics and needs are defined as user profile [1]-[4]. The practice of gathering, organizing and interpreting the user profile information is called user profiling [5][6].

The main challenge in user profiling is the generation of initial user profile for a new user and the continuous update of the profile information to adapt their changing preferences, interests and needs. In literature two fundamental user profiling methods have been proposed to build and handle user profiles. These are the content-based and the collaborative methods (see Chapter 2, Subsection 2.2.1. and Subsection 2.2.2.).

The literature review carried out in this thesis on user profiling shows the wide use of collaborative and content-based methods for the personalization in various applications (i.e. personalized handheld services, personalized web services, personalized television services) (see Chapter 2, Subsection 2.2.4.). This review also reflects the importance of user profiling features such as ratings, items, keywords and simple demographics to represent each user (see Chapter 2, Subsection 2.2.4.). Although the conventional way of profiling works well for specific services, it lacks in representing the multidimensionality of the user profiles accurately (see Chapter 2, Subsection 2.2.5.). For example, user profiles that reflect the ratings which were given to music videos cannot be used to recommend books for the same user. This constraint motivated the need to conduct more advance profiling to build a more comprehensive profiles to describe user's interest, preferences and demographics that can be used by various third party service providers for different service personalization. To address this problem, the author investigated various classification and clustering algorithms for user profiling and evaluated their performances with

different user profile datasets (see Chapter 3). The experiments presented in this thesis were conducted by using user profile datasets that reflect the user's personal information, preferences and interests.

From the given information, simulation results and comparisons of the algorithms, the utilization of the Instance Based Learner (IBL) classification algorithm for user profiling is preferred to be the main focus for the rest of the research work (see Chapter 3, Subsection 3.3.1. and Section 3.5.). This is because, compared to the other algorithms, IBL has the following properties;

- processes instances incrementally,
- is fast and robust,
- can represent probabilistic and overlapping concepts,
- assumes that the similar instances have similar classification that is similar to the concept of the user profiling where similar users with similar profiles share similar personal interest and preferences, and
- has potential to be improved to give better performance for user profiling.

However, IBL does not consider the relevancy of the user profile information during the user profiling. To be able to use the multidimensional profiles effectively, feature weighting should be taken into account. The utilization of feature weighting is therefore essential for accurate user profiling. This is mainly because the relevancy of all information contained within the user profile is not the same for different service personalization. For example, user's book interest information may not be as relevant as the income information of the user for personalized restaurant recommendations. Using weights to make the

distinction between relevant and irrelevant information could provide a solution for this problem. Considering this possible solution, a novel clustering algorithm for the user profiling, namely Multi- Dimensional Clustering (MDC), has been proposed in this thesis (see Chapter 5).

The MDC is a modified version of the IBL algorithm. In IBL every feature has an equal effect on the classification regardless of their relevancy. MDC differs from the IBL by assigning weights to feature values to distinguish the effect of the features on clustering (see Chapter 5, Section 5.2.). For the MDC's feature value weighting, feature weighting methods (Wrapper and Filter methods), which balance the effect of relevant and irrelevant user information during classification, are investigated (see Chapter 4). Following this investigation, three feature weighting methods have been proposed for the MDC. These methods are; MDC weight method by Cross Clustering (MDC-CC), MDC weight method by Balanced Clustering (MDC-BC) and MDC weight method by changing the Lower-limit to Zero (MDC-LZ) (see Chapter 5, Subsection 5.2.1., Subsection 5.2.2. and Subsection 5.2.3.).

Simulations were carried out with all of the proposed weighted MDC algorithms in addition to IBL and existing weighted and non-weighted IBL algorithms (i.e. K-Star and Locally Weighted Learning (LWL)) (see Chapter 5, Subsection 5.3.2.). The general conclusion, based on the simulations and evaluations, is that MDC-LZ algorithm produces better clustering accuracy performance for user profiling compared to all other algorithms. Hence, with the MDC-LZ, the author achieved the aim of proposing and implementing a weighted clustering algorithm that improves the accuracy of existing methods of user profiling and can perform

multidimensional user profiling that can be used for the personalization of different services.

6.2. Conclusions

In this thesis we investigated existing user profiling methods, classification and clustering algorithms, and feature weighting methods for the user profiling. A novel weighted clustering algorithm named Multi-Dimensional Clustering (MDC), using a combination of classification and clustering for the purpose of improving the accuracy of the existing methods of user profiling, was proposed and evaluated. MDC is a modified version of the Instance Based Learner (IBL) and it assigns weights to the feature values and performs clustering of the users based on the weighted distances.

In addition, three novel weighting methods for the MDC were proposed. These methods namely CC, BC and LZ, were used to improve the clustering accuracy of the new algorithm. The proposed algorithm, with each of the weighting method, was implemented on JAVA and MATLAB platforms and analysed using computer simulations on various user profile datasets. The simulation results indicated that each of the three weighted versions of MDC (MDC-CC, MDC-BC and MDC-LZ) improved the accuracy of IBL. The MDC-LZ performed better than MDC-CC and MDC-BC by reducing the error rate of IBL by as much as 10%.

Overall, this research was successfully carried out and all original aims and objectives have been achieved.

Personalization of services can improve quality of service and achieve optimum user satisfaction. Demand on personalized services will be much higher in the

future. The success of these services relies on how well the user requirements are reflected on the user profile and the services. Therefore an efficient user profiling method can provide accurate user profiles for different service personalization.

In this thesis a systematic study of the user profiling was carried out, with the following main conclusions for each chapter.

In *Chapter 2* the fundamentals of the user profiling are presented, starting by defining the user profile. A comparison of user profile types was carried out, and the advantages and disadvantages of each category were listed. The terminology used throughout this thesis was defined. In addition, the significance of the user profiling for a number of technological methods and applications were discussed in detailed. Various user profiling methods: the collaborative, content-based and the hybrid were described. A comparison of these methods was carried out, addressing the main techniques, advantages and disadvantages of each user profiling method. Some of the research works and standards published for user profiling were given. Finally, two popular applications were described as examples of user profiling methods.

In *Chapter 3* classification and clustering for user profiling has been discussed in detail. The clustering methods studied in this chapter were: Hierarchical clustering, Partitional clustering and Density-based clustering. A comparison of these methods was carried out and time and space complexity, clustering type, cluster type, data objects and dataset factors of each method were listed. Moreover, classification algorithms such as Decision Trees (DTs), Nearest Neighbour (NN) Classifiers, Support Vector Machine (SVM), and Bayesian

Classification were also presented. Some of the research works about the classification algorithms were also described.

Chapter 3 also evaluated the most popular algorithms of classification, such as LBR, NBTree, NB, BN and ID3. The classification accuracy performance of these classifiers on user profile data was presented. All simulations were performed in the Weiko Environment for Knowledge Analysis (WEKA) machine learning platform. The simulations were conducted using a variety of user profile datasets that represents the user's personal information (demographic data), interest and preference information. A University of California Irvine (UCI) adult dataset was used and modified to provide demographic profile information. Simulations conducted on IBL, BN, NB and LBR carried out with two different datasets containing 20 instances and 10 and 18 attributes. The simulation results showed that the BN classifier achieved the worst classification accuracy at 85% and 80% in each dataset. Furthermore, the classification accuracy of NB and IBL classifiers was 95%. Hence, the simulation results on both datasets showed that NB and IBL performed better in comparison to BN and LBR classifiers on small datasets.

Simulations on user profile dataset with 1000 instances and 18 attributes were carried out to obtain the classification accuracy of NB, IBL, SimpleCART, NBTree, ID3, J48 and SMO. Simulation results showed that the NBTree classifier achieved the best classification accuracy, at 90.20%, but has the highest computational requirement to build the classification model. Moreover, SimpleCart and J48 classifiers were achieved classification accuracy of 89.50% and 89.80% respectively. The results also showed that the worst classification accuracy was achieved by the ID3 at approximately 74.30%.

In *Chapter 4* the feature weighting methods Filter and Wrapper methods were presented. The Filter methods rely on the probabilistic distribution of the clusters and/or the features. Therefore, the statistics of the components are considered during the weight assignment when these methods are preferred. In this chapter, Filter methods such as Conditional probabilities, Class Projection and Mutual Information were discussed.

The Wrapper methods, also called Feedback methods, adaptively update the feature weights depending on the selected algorithm. A feedback is required to run the Wrapper methods, which feeds the decision of the classifier back to the algorithm. The algorithm then increments or decrements the corresponding feature weights accordingly. Chapter 4 also discussed the Incremental Hill-climbers and Continues Optimizer Wrapper methods.

The disadvantage of wrapper methods is that they are costly and time consuming with the high dimensional data. However, the advantage of these methods over the filter methods is the feedback mechanism. Detailed information on the advantages and disadvantages of both methods were also given in Chapter 4. Finally, two of the better known algorithms of each feature weighting method are defined and utilization of Filter and Wrapper models for user profiling was discussed.

In *Chapter 5*, a novel clustering algorithm named Multi-Dimensional Clustering (MDC) was proposed and evaluated for user profiling. MDC is a modified version of IBL and it assigns weights to feature values and provides clustering of the users based on the weighted distances. IBL is a comprehensive form of the Nearest Neighbour (NN) algorithm and it is suitable for user profiling as users

with similar profiles are likely to share similar personal interests and preferences.

Three feature weighting methods were proposed for the MDC as listed below:

1. Cross Clustering (CC)
2. Balanced Clustering (BC)
3. Lower-limit to Zero (LZ)

The CC method makes use of the probabilistic distribution of the feature values among the clusters to calculate the weight values for MDC. BC takes also into account the distribution of clusters along with the concept that has been introduced by the CC method. LZ completely removes the effect of irrelevant feature values while boosting the effect of relevant feature values on clustering.

The MDC-CC, MDC-BC, MDC-LZ and IBL were simulated with various user profile datasets to validate their performances. The evaluation of the results were done based on the clustering accuracy and error percentage.

Two sets of user profile dataset were used for the simulations. These included a training dataset that has 10000 instances and a test dataset that included 1000 instances. The first simulations were conducted on IBL to show the improvement in error rate with different dimensions of the user profile data. The simulation results showed that the error rate of the IBL is the lowest (35%) when all dimensions of the user profile, including demographic profile, interest profile and preference profile data has been used. The second simulations were carried out with the MDC-CC, MDC-BC and MDC-LZ. The simulations results indicated that each of the proposed MDC versions reduced the error rate of the

IBL. In addition, it is shown that the MDC-LZ performs better than MDC-CC and MDC-BC by reducing the error rate of IBL up to 10%. The performance of the IBL and MDC-LZ was also tested over test datasets of different sizes. The results showed that the performance of these algorithms stays almost the same even if different sets of test data were utilized.

Utilization of the PCF weighting for the IBL was investigated and evaluated. The PCF's variety of weight values was found to be greater than the proposed MDC weighting methods. However, it was proven that the PCF method was not capable to achieve correct clustering. Two straightforward modifications were discussed to improve clustering performance of PCF. Although these modifications overcome the PCF's issue on accurate clustering, the simulations results were not promising to enable the use of PCF for user profiling. The simulation results indicated that the error rate for PCF is up to 44% and the decrease in the error rate is not more than 10%. Hence, it was concluded that the PCF is less effective when it is used for the purpose of multi-dimensional clustering for user profiling.

Additional simulations were carried out with weighted and non-weighted IBL algorithms namely IBK, dw-IBK (1/d), dw-IBK(1-d), KStar and LWL. The results of these simulations were presented in a table that compares the MDC-LZ MDC-CC, and MDC-BC with IBK, dw-IBK (1/d), dw-IBK(1-d), KStar and LWL in terms of 1) the distance metric, 2) number of neighbours involved in classification, 3) weighting function, 4) how the label prediction is done and finally 5) error rate.

The main conclusion was that the MDC-LZ algorithm produces better clustering accuracy performance compared to all algorithms.

The last section of Chapter 5 aims to show how the MDC algorithm could be used for the user profiling to improve personalized service provisioning in mobile environments. For this purpose a real life scenario was implemented as a Java Mobile Application (Java ME) on NetBeans IDE 7.1.

6.3. Future Works

The following topics are suggested for future work:

- The use of weighting methods to distinguish the relevant and irrelevant features is new to user profiling. The studies on multi-dimensional weighting methods can further be modified to other classification/clustering methods given in Chapter 3. This has been shown to work well along with user profile data.

- Due to their algorithmic limitations, the Per-Category Feature weighting method could not be adapted to user profiling. Although it has presented clearly why these methods do not work for the given system, it would be of interest to modify the structures of this weighting method in order to make use of them in user profiling.

Finally, there is a limited number of works in the literature studying user profiling. Hence, the subject area of this thesis can be easily adapted to new research studies. Although this research study has mainly focused on clustering and classification for use profiling, it is possible to incorporate other concepts as

presented in Chapter 2. For instance, explicit and implicit profiles, improvement of collaborative and the content-based methods are individually areas worthy of further research.

References

[1] G. Araniti, P. D. Meo, A. Iera and D. Ursino (2003). Adaptive controlling the QoS of multimedia wireless applications through user profiling techniques, *IEEE Journal on selected areas in communication*, 21(10), pp. 1546-1556.

[2] T. Kuflik and P. Shoval (2000). Generation of user profiles for information filtering-research agenda, *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 313-315.

[3] M. J. Martin-Bautista, D. H. Kraft, M. A. Vila, J. Chen and J. Cruz (2002). User profiles and fuzzy logic for web retrieval issues, *Soft Computing (Focus)*, 15(3-4), pp. 365-372.

[4] European Telecommunications Standards Institute (ETSI) (2005). Human Factors (HF); User Profile Management, pp.1-100, Available: http://www.etsi.org/deliver/etsi_eg/202300_202399/202325/01.01.01_60/eg_202325v010101p.pdf

[5] Oxford dictionaries online (2010) *Oxford University Press*. Available: http://oxforddictionaries.com/definition/profiling?q=profiling

[6] S. Henczel (2004). Creating user profiles to improve information quality, *Factiva*, 28(3), p. 30.

[7] M. R. Lopez, A. B. B. Martinez, A. Peleteiro, F. A. M. Fonte and J. C. Burguillo (2011). moreTourism:mobile recommendations for tourism, *IEEE International Conference on Consumer Electronics*, pp. 347-348.

[8] Y. B. Fernandez, M. L. Nores, J. J. P. Arias, J. G. Duque, M.I.M. Vicente (2011). TripFromTV+:Exploiting social networks to arrange cut-price touristic packages, *IEEE International Conference on Costumer electronics*, pp. 223-224.

[9] C. K. Georgiadis and S. H. Stergiopoulou (2008). Mobile commerce applications development: implementing personalized services, *International Conference on Mobile Business*, pp. 201-210.

[10] W. Woerndl, C. Scheuller and R. Wojtec (2007). A hybrid recommender system for context-aware recommendations of mobile applications, *IEEE International Conference on Data Engineering Workshop*, pp. 871-878.

[11] H. Jeon, T. Kim and J. Choi (2008). Mobile semantic search personal preference filtering, *International Conference on Networked Computing and Advanced Information Management*, pp. 531-534.

[12] M. Khosrowpour (2005). "Encyclopaedia of information science and technology", Electron. Book, Hershey, PA Idea Group Reference, pp. 2063-2067.

[13] C. Gena (2005). Methods and techniques for the evaluation of user-adaptive systems*, The Knowledge Engineering*, 20(1), pp. 1-37.

[14] D. Poo, B. Chng and J. M. Goh (2003). A hybrid approach for user profiling*, Annual Hawaii International Conference on System Sciences, 4(4),* pp. 1-9*.*

[15] B. Dean (2006). Daily market movers digest stock alerts, *Factiva*, p. 1.

[16] Z. Xujuan, S.T.Wu, Y. Li, Y.Xu, R.Y.K. Lau and B.D. Bruza (2006). Utilizing search intent in topic ontology based user profile for web miming, *IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 558-564.

[17] J. Blom (2000). Personalization - a taxonomy, *Conference on Human Factors in Computing Systems*, pp. 313-314.

[18] I. Jorstad, D. V. Thanh and S. Dustdar (2004). Personalization of Future Mobile Services, *International Conference on Intelligence in Service Delivery Networks*.

[19] I. Jorstad, D. V. Thanh and S. Dustdar (2005). The personalization of mobile services, *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, 4, pp. 59-65.

[20] H. Stormer (2004). Personalized websites for mobile devices using dynamic cascading style sheets, *International Conference on Advances in Mobile Multimedia,* pp. 351-360.

[21] E. Lillevold and J. Noll (2004). Personalization in telecom business, European Institute for Research and Strategic Studies in Telecommunications Available: http://archive.eurescom.eu/message/messageMar2004/Personalisation_in_telecom_business.asp.

[22] R. Guarneri, A.M. Sollund, D. Marston, E. Fossbak, B. Berntsen, G. Nygreen, G. Gylterud, R. Bars and A. Kerdraon (2004). Report of state of the art in personalisation, Common Framework, pp. 1-59, Available: http://www.ist-eperspace.org/deliverables/D5.1.pdf

[23] P. S. Yu (1999). Data mining and personalization technologies, *International conference on database systems for advance applications*, pp. 6-3.

[24] D. Kelly and J. Teevan (2003). Implicit feedback for inferring user preference: a bibliography, *ACM Special Interest Group on Information Retrieval (SIGIR) forum*, 37(2), pp. 18-28.

[25] J. Wang (2006). “Encyclopaedia of data warehousing and mining”, Electron. Book, Hershey, PA Information Science Reference, p.144.

[26] A. B. Tucker (2004). “Computer science handbook”, Electron. Book, 2nd Edition, Boca Raton, Fla CRC Press, pp.75-10.

[27] L. Rivero, J.H. Doorn and F. Viviana (2006). “Encyclopaedia of database technologies and applications”, Electron. Book, Hershey, PA Information Science Reference, p.318.

[28] A.L. Symeanidis and P.A. Mitkas (2005). “Agent intelligence through data mining multiagent systems, artificial societies and simulated organizations; 14”, Electron. Book, New York: Springer Science and Business Media, p.27.

[29] M. Khosrowpour (2006). “Encyclopaedia of ecommerce, e-governments, and mobile commerce”, Electron. Book, Hershey, PA Information science Reference, pp.118-123.

[30] S. Steward and J. Davies (1997). User profiling techniques: a critical review, *British Computer Society*, *BCS-IRSG Annual Colloquium on IR Research,* pp.1-22.

[31] E. J. Neuhold (2003). Personalization and user profiling & recommender systems, *WI/IM Information Management Proseminar*, pp. 1-25.

[32] H. Luo, C. Niu, R. Shen and C. Ullrich (2008). A collaborative filtering framework based on both local user similarity and global user similarity, *Springer Computer Science Machine Learning*, 72(3), pp. 231-245.

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

[34] X. Su and T. M. Khoshgoftaar (2009). A survey of collaborative filtering techniques, *Advances in Artificial Intelligence*, p.p. 1-19.

[35] D. Godoy and A. Amandi (2005). User profiling for web page filtering, *IEEE internet computing*, 9(4), pp. 56-64.

[36] R. Carson and M. Hearst (1998). Term weighting and ranking algorithms, Lecture Notes: Information Organization and Retrieval, University of California, Available: http://www2.sims.berkeley.edu/courses/is202/f98/Lecture17/sld001.htm.

[37] C. Biancalana, F. Gasparatti, A. Micarelli and G. Sansonetti (2011). Social tagging for personalized location-based services, International Workshop on Social Recommender Systems, pp.1-9

[38] J. Park, S. J. Lee, S. J. Lee, K. Kim, B. S. Chung and Y. K. Lee (2011). Online video recommendation through tag-cloud aggregation, *IEEE Multimedia*, 18(1), pp. 78-87.

[39] C. A. Yeung, N. Gibbins and N. Shadbolt (2008). A study of user profile generation from folksonomies, *Workshop on Social Web and Knowledge Management,* pp. 1-8.

[40] K. Lakiotaki, N. F. Matsatsinis and A. Tsoukias (2011). Multicriteria user modelling in recommender systems, *IEEE Intelligence Systems*, 26 (2), pp. 64-76.

[41] R. V. Meteren and M. V. Someren (2000). Using content-based filtering for recommendation, *Workshop on Machine Learning in the New Information Age*, pp. 312-321.

[42] Y.W. Park and E.S. Lee (1998). A new generation method of a user profile for information filtering on the internet, *International Conference on Information Networking*, pp. 261-264.

[43] G. Specht and T. Kahabka (2000). Information filtering and personalization in databases using gaussian curves, *International Symposium on Database Engineering and Applications*, pp. 16-24.

[44] A. B. B. Martinez, M. R. Lopez, E. C. Mantenegro, J. C. Burguillo, F. A. M. Fonte and A. Peleteiro (2010). A hybrid content-based and item-based collaborative filtering to recommend TV programs enhanced with singular value decomposition, *Elsevier Information Sciences: an International Journal*, 180(22), pp. 4290-4311.

[45] Z. S. Shibeshi, S. Ndakunda, A. Terzoli and K. Brandshow (2011). Delivering a personalized video service using IPTV, *International Conference on Advanced Communication Technology*, pp. 1489-1494.

[46] M. Kodialam, T.V. Lakshman, S. Mukherjee and L. Wang (2011). Online scheduling of targeted advertisements for IPTV, *IEEE/ACM Transactions on Networking*, 19(6), pp.1825-1834.

[47] T. Pessemier, T. Deryckere, K. Vanhecke and L. Martens (2008). Proposed architecture and algorithm for personalized advertising on iDTV and mobile devices, *IEEE Transactions on Consumer Electronics*, 54(2), pp. 709-713.

[48] Amazon.com (2012). Available: www.amazon.co.uk

[49] Yahoo! (2012). Available: www.music.yahoo.com

[50] I. Jorstad and D. V. Thanh (2006). Service personalization in mobile heterogeneous environments, *Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services,* pp.70-75.

[51] Vodafone (2001). Work item description: the 3GPP generic user profile, pp. 1-5, Available: http://www.3gpp.org/ftp/Information/WI_Sheet/_Archive/SP-010548.pdf.

[52] Wikipedia (2007). Data Mining [Online] Available : http://en.wikipedia.org/wiki/Data_mining

[53] A.L. Symeanidis and P.A. Mitkas (2005). “Agent intelligence through data mining multiagent systems, artificial societies and simulated organizations; 14”, Electron. Book, New York: Springer Science and Business Media, pp.21-23, 27-28.

[54] M. Sushmita and A. Tinku (2003). “Multimedia, soft computing and bioinformatics” Electron. Book, Hoboken, N.J. John Wiley and Sons, Inc.(US), pp 18-19.

[55] F.V. Jensen (1993). “Introduction to Bayesian networks”. Denmark, Hugin Expert A/S.

[56] L. Jiang and Y. Guo (2005). Learning lazy Naïve Bayesian classifier for ranking, *IEEE Conference on Tools with Artificial intelligence*, pp. 412-416.

[57] Z. Wang and G. I. Webb (2002). Comparison of lazy Bayesian rule and tree-augmented Bayesian learning, *IEEE conference on Data Mining*, pp. 490-497.

[58] Z. Shi, Y. Huang and S. Zhang (2005). Fisher score based naive Bayesian classifier, *IEEE International Conference on Neural Networks and Brain*, pp. 1616-1621.

[59] J.M. Pena, J. A. Lozano and P. Larrañaga (1999). Learning Bayesian networks for clustering by means of constructive Induction, Pattern Recognition Letters, 20(11-13), pp. 1219-1230.

[60] Z. Xie and Q. Zhang (2004). A study of selective neighbourhood-based naïve Bayes for efficient lazy learning, *IEEE International Conference on Tools with Artificial Intelligence*, pp 758-760.

[61] G. Santafe, J.A. Loranzo and P. Larranaga (2006). Bayesian model averaging of naive bayes for clustering, *IEEE Transactions on Systems, Man, and Cybernetics*, 36(5), pp. 1149 -1161.

[62] V. P. Bresferean (2007). Analysis and predictions on student's behaviour using decision trees in WEKA environment, *IEEE International Conference on Information Technology Interfaces*, pp.51-56.

[63] M.J.A. Berry and G. Linoff (2004). "Data mining techniques: for marketing, sales and customer relationship management", 2nd Edition, Electron. Book, Indianapolis John Wiley and Sons. Inc. (US), pp.11, 165-167.

[64] J. R. Quinlan (1990). Decision trees and decisionmaking, *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2), pp. 339-346.

[65] L. Liu, Z. Liz and H. He (2008). The research of decision support vector machine in web information classification, IEEE *International Conference on Computer Supported Cooperative Work in Design,* pp. 196-200.

[66] K.P. Bennett and J. A. Blue (1998). Support vector machine approach to decision trees, IEEE *International Conference on Neural Networks*, pp. 2396-2401.

[67] T. Segaran (2007). "Programming collective intelligence" Electron. Book, O'Reilly Media, Inc. pp. 216.

[68] D. W. Aha, D. Kibler and M. K. Albert (1991). Instance-based learning algorithms, *Machine Learning Journal*, 1(6), pp. 37-66.

[69] I. H. Witten and E. Frank (2005). "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco.

[70] M. M. Irene (1999). Clustering Methodology, Available : http://www.cse.iitb.ac.in/dbms/Data/Courses/CS632/1999/clustering/node1.html

[71] Hierarchical Clustering (2008). Available: http://fedc.wiwi.hu-berlin.de/xplore/tutorials/xaghtmlnode53.html

[72] A tutorial on clustering algorithms (2000). Available: http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/hierarchical.html

[73] A.K. Jain, A. Topchy, M. H. C. Law and J. M. Buhmann (2004). Landscape of Clustering Algorithms, *International Conference on Pattern Recognition*, 1, pp. 260-263.

[74] A.K. Jain and R.C. Dubes (1998). "Algorithms for clustering data". , 1st Edition, Prentice-Hall Advanced Reference Series, Prentice Hall, Inc., Upper Saddle River, NJ, pp.1-304.

[75] A. A. Shah (2007). "User profiling based on clustering techniques" (Master of Science thesis), University of Westminster, Department of Electronics, Communication and Software Engineering, pp. 34-36.

[76] A.K. Jain, M.N. Murty and P.J. Flynn (1999). Data clustering-review, *ACM computing surveys*, 31(3), pp.60.

[77] M. Ester, H.P. Kriegel, J. Sander and X. Xu (1996). A density-based algorithm for discovering clusters in large spatial databases with noise, *International Conference on Knowledge Discovery and Data Mining*, pp. 226-231.

[78] M. Panda and R. M. Patra (2008). A comparative study of data mining algorithms for network intrusion detection, *International Conference on Emerging Trends in Engineering and Technology,* pp.504-507.

[79] H. Zhang and J. Su (2004). Naive Bayesian classifiers for ranking, *European Conference on Machine Learning,* pp 1-12.

[80] A. Asuncion and D.J. Newman (2007). UCI Machine Learning Repository Irvine, CA: University of California, School of Information and Computer Science.

[81] F. Provost and P. Domingos (2003). Tree induction for probability based ranking, *Machine Learning*, 52(3), pp. 199-215.

[82] J. Huang and C. X. Ling (2005). Using AUC and accuracy in evaluating learning algorithms, *IEEE Transactions on Knowledge and Data Engineering*, 17(3), pp. 299-310.

[83] D. Irani, S. Webb and C. Pu (2010). Study of static classification of social spam profiles in MySpace, *International Conference on Weblogs and Social Media,* pp. 82-89.

[84] A. Cufoglu, M. Lohi and K. Madani (2008). A comparative study of selected classification accuracy in user profiling, *International Conference on Machine Learning and Applications*, pp.787-791.

[85] A. Cufoglu, M. Lohi and K. Madani (2008). classification accuracy performance of Naïve Bayesian (NB), Bayesian Networks (BN), Lazy Learning of Bayesian Rules (LBR) and Instance-Based Learner (IB1) - comparative study, *International Conference on Computer Engineering and Systems*, pp. 210-215.

[86] A. Cufoglu, M. Lohi and K. Madani (2009). A comparative study of selected classifiers with classification accuracy in user profiling, *World*

Congress on Computer Science and Information Engineering, pp.708-712.

[87] The university of Waikato, Weka (1993-2009). Publications Available from: http://www.cs.waikato.ac.nz/~ml/publications.html

[88] D. Xhemali, C. J. Hinde and R. G. Stone (2009). Naïve Bayes vs. decision trees vs. neural networks in the classification of training web sites, *International Journal of Computer Science*, 4(1), pp. 16-23.

[89] J. G. Clear and L. E. Trigg (1995). K*: An instance-based learner using an entropic distance measure, *International Conference on Machine Learning*, pp. 108-114.

[90] G. Demiroz and H. A. Guvenir (1996). Genetic algorithms to learn feature weights for nearest neighbour algorithm, *Belgian-Dutch Conference on Machine Learning*, pp. 117-126.

[91] G. John, R. Kohavi and K. Pfleger (1994). Irrelevant features and the subset selection problem, *international Machine Learning Conference*, pp.121-129.

[92] D. Wettschereck and D.W. Aha (1995). Weighting Features, *International Conference on Case-Based Reasoning Research and Development,* pp.347-358.

[93] X. Tong, P. Ozturk and M. Gu (2004). Dynamic feature weighting in nearest neighbour classifier, *International Conference on Machine Learning and Cybernetics*, 4, pp. 2406-2411.

[94] O. Söder (2008). kNN classifiers: Feature weighting. Available: http://www.fon.hum.uva.nl/praat/manual/kNN_classifiers_1_1_1__Feature_weighting.html

[95] J. Liu and G. Wang (2010). Hybrid feature selection method for datasets of thousands of variables, *International Conference on Advanced Computer Control,* 2, pp. 288-291.

[96] H. Yuan, S. S. Tseng, W. Gangshan and Z. Fuyan (1999). A two-phase feature selection methods using both filter and wrapper, IEEE conference on Systems, Man and Cybernetics, 2, pp.132-136.

[97] K. Kira and L.A. Rendell (1992). A practical approach to feature selection, International Conference on Machine Learning, pp. 249-256.

[98] R. Kohavi and G.H. John (1997). Wrappers for feature subset selection, *Elsevier Science B.V.*, 97 (1-2), pp. 273-324.

[99] R. H. Creecy, B.M. Masand, S. J. Smith and D. L. Waltz (1992). Trading MIPS and memory for knowledge engineering, *Communications of the ACM*, 35 (8), pp.48-63.

[100] C. Stanfill and D. Waltz (1986). Towards memory based reasoning, *Communications of the Association for Computing Machinery*, 29 (12), 1213-1228.

[101] V. Pekar, M. Krkoska and S. Staab (2004). Feature weighting for co-occurrence-based classification of words, *international conference on Computational Linguistics,* pp. 799-805.

[102] D.W. Aha (1990). "A study of instance-based algorithms for supervised learning tasks: mathematical, empirical, and psychological evaluations "(Doctoral thesis), Irvine, CA:University of California, Department of Information and Computer Science, pp. 1-406.

[103] S. Salzberg (1991). A nearest hyperrectangle learning method, *Journal of Machine Learning*, 6(3), pp.251-276.

[104] T. Mohri and H. Tanaka (1994). An optimal weighting criterion of case indexing for both numeric and symbolic attributes, *Workshop on Case-Based Reasoning*, pp. 123-127.

[105] D. Wettschereck, D.W. Aha and T. Mohri (1997). A Review and Empirical Evaluation of Feature Weighting Methods for a Class of Lazy Learning Algorithms, *Transactions on Artificial Intelligence*, 11(1-5), pp.1-37.

[106] O. Gomez, E F. Morales and J. A. Gonzales (2007). Weighted instance-based learning using representative intervals, *Mexican International Conference on Advances in Artificial Intelligence*, pp. 420-430.

[107] I. H. Witten, E. Frank and M. A. Hall (2011). "Data mining practical machine learning tools and techniques" 3rd Edition, Morgan Kaufmann, USA pp. 472-550.

[108] T. Searan (2007). "Programming collective intelligence", 1st Edition, Electron. Book, O'Reilly Media, Inc.

[109] C. G. Atkeson, A. W. Moore and S. Schaal (1997). Locally weighted learning, *Artificial Intelligence*, pp.11-73.

[110] K. F. Yeung and Y. Yang (2010). A proactive personalized mobile news recommendation system, *IEEE Developments in E-systems Engineering*, pp.207-212.

[111] B. Mrohs and S. Steglich (2005). Architectures of future services and applications for mobile user, *IEEE Symposium on Applications and the Internet Workshops,* pp. 128-131.

[112] S. Loeb and E. Panagos (2011). Information filtering and personalization: context, serendipity and group profile effects, *Annual*

IEEE Consumer Communications and Networking Conference-Emerging and Innovative Consumer Technologies and Applications, pp. 393-398.

[113] T. D. Pessemier, T. Deryckere, K. Vanhecke and L. Martens (2008). Proposed architecture and algorithm for personalized advertising on iDTV and mobile devices, *IEEE Transactions on Consumer Electronics*, 54(2), pp.709-713.

[114] M. N. Gasson, E. Kosta, D. Royer, M. Meints and K. Warwick (2011) Normality mining: privacy implications of behavioural profiles drawn from GPS enabled mobile phones, *IEEE transactions on systems, man, and cybernetics-Part C: Applications and reviews*, 41(2), pp. 251-261.

Appendix A

List of Publications

[1] A. Cufoglu, M. Lohi and K. Madani (2008). A Comparative Study of Selected Classification Accuracy in User Profiling, *7th International Conference on Machine Learning and Applications*, pp.787-791.

[2] A. Cufoglu, M. Lohi and K. Madani (2008). Classification accuracy performance of Naïve Bayesian (NB), Bayesian Networks (BN), Lazy Learning of Bayesian Rules (LBR) and Instance-Based Learner (IB1) - comparative study, *International Conference on Computer Engineering and Systems*, pp. 210-215.

[3] A. Cufoglu, M. Lohi and K. Madani (2009). A Comparative Study of Selected Classifiers with Classification Accuracy in User Profiling, *2009 World Congress on Computer Science and Information Engineering,* pp.708-712.

[4] A. Cufoglu, M. Lohi and C. Everiss (2012). Weighted Instance Based Learner (WIBL) for User Profiling, *IEEE 10th Jubilee International Symposium on Applied Machine Intelligence and Informatics*, pp. 201-205.

[5] A. Cufoglu, M. Lohi and C. Everiss (2012). Personalized Mobile Services Using Weighted Instance Based Learner for User Profiling, *5th International Conference on Advances in Human-oriented and Personalized Mechanisms*, Technologies and Services, pp.128-132.

[6] A. Cufoglu, M. Lohi and C. Everiss (2013). User Profiling - A New Approach, submitted to journal.

[7] A. Cufoglu, M. Lohi and C. Everiss (2013). Clustering Algorithms and Weighted Instance Based Learner for User Profiling, submitted to conference.