The Use of Items Personality Profiles in Recommender Systems

By:

Haifa Alharthi

A thesis submitted to the
Faculty of Graduate and Postdoctoral studies
In partial fulfillment of the requirements for the degree of
Masters in Computer Science

Ottawa-Carleton Institute for Computer Science
School of Electrical Engineering and Computer Science

University of Ottawa

Abstract

Due to the growth of online shopping and services, various types of products can be recommended to an individual. After reviewing the current methods for cross-domain recommendations, we believe that there is a need to make different types of recommendations by relying on a common base, and that it is better to depend on a target customer's information when building the base, because the customer is the one common element in all the purchases. Therefore, we suggest a recommender system (RS) that develops a personality profile for each product, and represents items by an aggregated vector of personality features of the people who have liked the items. We investigate two ways to build personality profiles for items (IPPs). The first way is called average-based IPPs, which represents each item with five attributes that reflect the average Big Five Personality values of the users who like it. The second way is named proportion-based IPPs, which consists of 15 attributes that aggregate the number of fans who have high, average and low Big Five values. The system functions like an item-based collaborative filtering recommender; that is, it recommends items similar to those the user liked. Our system demonstrates the highest recommendation quality in providing cross-domain recommendations, compared to traditional item-based collaborative filtering systems and content-based recommenders.

Acknowledgements

All praise and thanks are only for God, the One who, by His blessing and favor, good works are accomplished. Then, my thanks are for my beloved family especially Khalid who without their continuous love and support I would not have reached this point. In addition, thanks to my supervisor, Dr. Thomas T. Tran, who encouraged me to work on what I am passionate about and guided me through the way. Thanks to all professors in uOttawa, who taught me what I needed to achieve this work. Thanks to my friends and colleagues for their constant assistance and support.

Table of Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Definition
RSs	Recommender Systems
CF	Collaborative Filtering
CB	Content-based
BF	The Big Five personality Traits
E	Extraversion
A	Agreeableness
C	Conscientiousness
N	Neuroticism
O	Openness
NEO-PI-R	NEO Personality Inventory Revised
BFI	Big Five Inventory
TIPI	Ten-Item Personality Inventory
IPIP	International Personality Item Pool
IPPs	Items' personality profiles
IPPs-based CF	Collaborative Filtering Based on Items' Personality Profiles
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
ROC	Receiver Operating Characteristic
AUC/ ROC area	The Area Underneath the ROC Curve

1. Introduction

1.1. The Context

Currently, websites often overwhelm users with vast amounts of information, products and services, and experts believe the overload of products can lead to incorrect decisions [1]. Individuals want to surf the web efficiently and find their target easily, and recommender systems (RSs) can limit the options and suggest what best fits users' needs. Amazon, for example, recommends a multitude of items from many domains, including books, gadgets and other products. In such e-commerce websites, which cover a wide range of products, there is a need to provide recommendations across different categories which we refer to as domains. Even though many websites suggest products from multiple domains, their recommendations rely on the information found in each domain [2]. For example, when recommending a book, movie or song, an RS exploits the user's history of book purchases to suggest a book, and does not consider their purchases in the music or movie domains. This is known as single-domain recommendation, while transferring the knowledge of one domain to another is called cross-domain [3].

There are many reasons why it is important for RSs to draw on a user's purchase history in single domain (e.g., books) to make recommendations in other domains (e.g., movies). First, companies increase their revenues when they provide users with packages of various products, and users can get many products they want in one deal. Secondly, cross-domain recommendations help RSs avoid overspecialization, which is a problem when the system provides the target user with only items that are similar to those they already like. This can become tedious, as it is preferred to give users novel and serendipitous suggestions. Serendipitous recommendations are those that are surprising to the user, and novel recommendations are items which are new and unfamiliar to the target user. For instance, in case a user has preferred many movies for a specific actor, the user will not be surprised to be recommended with a new movie of the same actor (not serendipitous recommendation); however, the user might not know about the new movie (novel recommendation) [2] [4]. Cross-domain RS could surprise a user who is usually only interested in gadgets, by recommending a book they were not aware of; however, this can result in reduced recommendation accuracy. The accuracy means how accurate the predictions of a user's preferred items are. Thirdly, by exploiting the users' ratings in some domains to make recommendations in a not-yet-rated domain, cross-domain RSs can help address the users' cold start problem, which is the problem of making recommendations to new users who have not rated enough products. This scenario can occur when one system has multiple databases, or two businesses combine their data [2].

Traditional recommender systems are capable of making cross-recommendations, though products from these domains must be correlated in some way. The products might be correlated because they have similar descriptions, as when a user who has watched a lecture is recommended a research paper related to the lecture topic [3]. However, in some cases there are no commonalities in the content of the products (e.g., books are described by their genre, author, etc., and laptops by their CPU and screen size). As well, products from multiple domains could be suggested because they were rated by the same users; Amazon currently recommends items which have been rated by the community similarly to items preferred by the target user, the user receiving the suggestions [5]. Many other websites use similarity of the ratings of their community of users to personalize their services and products. However, users with rich ratings in one domain (e.g., books) do not always have many ratings in other domains (e.g., gadgets). Thus, the reliance on item descriptions or user ratings does not always allow cross-domain recommendations.

1.2. Motivation

This thesis is driven by the need for a new framework to make recommendations from multiple domains. The new system needs to create a unified representation for all products, regardless of their type, and this provides a basis for cross-domain recommendations. After reviewing the current methods for cross-domain recommendations, we found that they do not focus on the users' information. Instead, most of them focus on finding relationships between domains. In addition, users' information especially personality was not exploited by those systems to make recommendations. Since the user is the common element in all products a user purchases, the system should rely on the user's information especially personality. Depending on personality information can reveal important associations between users and products, and lead to accurate cross-domain recommendations. This belief is triggered by [6] and [7], which found a relationship between people's preferred genres and their psychological makeup.

1.3. Problem Statement

There is a need for RSs that can suggest items from multiple domains with high recommendation accuracy. Most research in the area of recommender systems focuses on making single-domain recommendations, while recommendations from multiple domains are not yet widely explored [2] [3]. Our research addresses the question of whether or not using personality information in making cross-domain recommendations can provide high recommendation accuracy. In this work, we capture the personality traits of fans of a product and use them to create the representation of that product. Then, we find items with a similar representation to recommend them to those fans. The user is the common element in all products a user purchases. Therefore, we think that a system which utilizes users' personality information would be useful for cross-domain recommendations.

1.4. Contributions

- This work addresses the issue of cross-domain recommendations which is not yet widely explored [2] [3].
- To the best of our knowledge, our research is the first that exploits the Big Five personality traits, a model to describe people's personality which most psychologists agree on [8] [9], in making cross-domain recommendations.
- We have collected a dataset of users' Big Five personality traits and their preferences in two domains.
- We propose a novel method (proportion-based IPPs) to represents products from different domains by using specific numbers of attributes that reflect the personality of the users who like the products.
- Our system demonstrates the highest recommendation quality in providing cross-domain recommendations, compared to traditional item-based collaborative filtering systems and content-based recommenders. In addition, our system performs as well as collaborative filtering in single-domain recommendations.
- This work has been exposed to the research community of the field and resulted in two technical papers: One paper was published in the Proceedings of the ACM RecSys-2014

Workshop on Recommender Systems for Television and Online Video [10], and another is currently submitted to Journal of Artificial Intelligence Research (JAIR) [11].

1.5. Thesis Organization

The organization of the thesis is as follows: Chapter 2 presents the background of recommender systems including collaborative filtering, content-based and other types of RSs. It also provides information about personality, and how it is measured and used in computer systems. Chapter 3 discusses related work in two sections: personality-based and cross-domain RSs. Chapter 4 explains the proposed system task and components in detail, then discusses how personality profiles for users and items are developed, and how to use the item personality profiles in making recommendations. Chapter 5 describes the dataset used in our experiments, the evaluation metrics and the results of the experiments. Finally, Chapter 6 summarizes the thesis, discusses the advantages and disadvantages of the system, and suggests future work.

2. Background

In this chapter, we provide a description of recommender systems and cross-domain recommendations, an explanation of two types of RSs, which are adopted in our work: collaborative filtering, content-based RSs, an overview of different kinds of RSs namely: demographic RSs, knowledge-based RSs, social RSs and hybrid RSs, and a brief background about personality, and how it is measured and used in computer systems.

2.1. Recommender Systems

Recommender systems (RSs) have been an active research area in recent years, and most university computer science departments now offer RS-related courses [1]. There are also conferences (e.g., ACM Recommender Systems (RecSys)) and journal issues (e.g., AI Communications (2008)) that address RS topics, and most well-known e-commerce websites (e.g., Amazon) have adopted RSs. The idea behind RSs is not new, as people have always asked acquaintances to recommend restaurants, movies, books, etc. However, RS suggestions are more complicated than simply what other consumers think of a product like in collaborative filtering systems. Sometimes, the focus is on the content of items or consumers' demographics [1]. The development of RSs had been aided by multidisciplinary sciences, including "…cognitive science, information retrieval, forecasting theories, approximation theory and management science" [12]. Basically, what recommender systems do is to predict which items (e.g., movies, cameras, books, etc.) suit a user who has not seen them, and suggest those items to the user. RSs effectively solve the problem of having too many products on the internet to choose from [1].

The process of making recommendations requires three components: items, users and user feedback on items. Things that RSs recommend to users are called items, regardless of if they are a service, a trip or any other product. The second component of a recommendation process is users. Users are the center of the system, as some researchers define RSs as software that develops and uses customers' profiles [1] . The information needed to build a user profile varies from system to system; RSs might exploit users' demographic information, ratings or personality. Feedback, the third component, is how users interact with items, and it can be collected explicitly or implicitly [1]. Explicit collection is when the users are asked their opinions about an item (e.g., to rate their experience), whereas implicit collection is when the system infers whether they like or dislike an item from their behavior (e.g., purchasing or viewing the product) [13]. Rating items can have many classes, such as 1 to 5 stars, two classes (like/dislike), or only one class (like) [1]. After receiving suggestions, a user may provide explicit or implicit feedback to indicate whether the user likes or dislikes the item; the system stores the user's opinion in a database, and uses it for future recommendations.

2.2. Cross-Domain Recommendations

There is no agreement among RSs researchers regarding what the term 'domain' refers to. Some describe a category of items (e.g., songs versus movies) as domains, while others considered genres as domains (e.g., romance versus action movies). In this work, we use the word domain to refer to category of items. The concept of cross-domain recommendation is defined as the use of information about items and users from one domain (e.g., movies) in another domain (e.g., books), in order to obtain high prediction quality [3]. The goal of an RS which provides cross-domain recommendations is to find correlations between what a user prefers in different domains, and use that knowledge to suggest items in the target domain. Cross-domain RSs can provide a user with a package of recommendations from various domains. For example, one RS can suggest Harry Potter movies, books, videogames and sound tracks to a user simultaneously [3].

2.3. Collaborative Filtering (CF)

Collaborative filtering is the most widely used approach of recommender systems [14]. CF systems exploit the available ratings of some users to predict and recommend items to another group of users. These systems are based on the assumption that two users who preferred similar items in the past will like similar items in the future. In general, CF predicts the preferences of users by exploiting a user-item matrix that has the ratings of m users $\{u_1, u_2, \ldots, u_m\}$ on n items $\{i_1, i_2, \ldots, i_n\}$. A user u_i has rated a list of items I_{ui}. If the user has not rated the item, a missing value is shown in the matrix [15]. In following sections, we discuss the overall advantages and disadvantages of CF, techniques used in CF systems, and cross-domain recommendations in CF systems.

2.3.1. Advantages and Disadvantages

CF does not require additional information about the content of items; it only needs the rating matrix in order to make suggestions. Another advantage is that CF is simple to implement. However, CF systems can have issues. Some businesses require large datasets, and this results in a sparse user-item rating matrix that makes it difficult for the system to make predictions, particularly for new users and items. When there are no previous ratings, or not enough ratings related to a user, the system cannot find similar users and this is called the cold start or new user problem. Also, the system cannot suggest a new item if no user has rated the item yet. The sparsity issue was addressed by attempting to reduce the dimensionality of the rating matrix, by removing non-important users and items [5]. However, this results in the loss of helpful information to the recommendation process, which can lead to reduced recommendation accuracy. Using hybrid RSs, which exploit external information about users or items, is a useful way to solve the sparsity problem. CF also suffers from having users, called ‘gray sheep’, with completely different tastes from the rest of the other users. However, humans also fail to predict gray sheep preferences, so it is not considered a serious problem in RSs. In addition, because CF systems rely on the ratings of users they are vulnerable to shilling attacks, which occur when some fake users rate specific products highly as a form of promotion, and give competing products low ratings. This may lead the system to recommend the promoted products to many users [15] [4]. It was determined that item-based CF is stronger than user-based CF when facing shilling attacks [16].

2.3.2. Collaborative Filtering Techniques

Two techniques are used in CF systems: model-based and memory-based CF. In model-based CF, machine learning algorithms such as Bayesian belief nets and clustering models are exploited to generate models that learn the pattern of users’ ratings and generate prediction of their future ratings. When users’ ratings are categorical the classification algorithms are applied whereas when the ratings are numerical regression methods are used. Model-based CF systems are able to deal with the sparsity and scalability problems. However, as model-based systems become scalable, the accuracy of recommendation decreases. In addition, developing the model can be an expensive process [15].

There are two approaches of memory-based CF algorithms: user-based and item-based [15]. In the former, it is assumed that two users with similar tastes or rating histories will rate items similarly in the future. Each user is represented as a vector of items’ ratings, which is called the user profile. The system finds users with similar profiles to the target user and exploits their ratings to predict the likeliness that the user likes a particular item [17]. In contrast, item-based CF system computes the similarity between two co-rated items, and the most similar items to those the target user has already preferred are recommended [18]. An illustrated example of both systems is shown in section 2.3.3. In our work, the k-nearest neighbor algorithm is adopted to make the item-based recommendations.

2.3.2.1. K-Nearest Neighbor Collaborative Filtering

It is a lazy algorithm since it puts the training dataset in memory and when it needs to make a prediction in an item-based CF, it searches for the items which have similar vectors to the item in the test dataset. Based on the target class (e.g., like/dislike) or the numeric ratings (e.g., 1-5 stars) of the similar items (neighbors), the prediction of the unrated items is made. The process of searching for the similar items involves using a similarity function [19]. We have used Euclidean distance. In an item-based CF, Euclidean distance is the distance between two items $X1=(x_{11},x_{12},:::x_{1n})$ and $X2=(x_{21},x_{22},:::,x_{2n})$ as in Eq. 1 [20].

Eq. 1

$$dist\ (X1,\ X2)= \sqrt{\sum_{i=1}^{n}(x_{1i}-x_{2i})^2}$$

2.3.3. Cross-Domain Recommendations in Collaborative Filtering

Recommendations for multiple domains can be done in a collaborative filtering system; an example of a CF that provides cross-domain recommendations is shown in Table 1. In a user-based CF as illustrated with red dotted-line, it is predicted that User 1 likes the book. That is inferred by comparing the column of User 1 to columns of the rest of the users and finding the most similar user profile (User 3). In item-based CF, in order to predict whether or not User1 likes the book, we search for the most similar row to the book's row. It is found that the movie is the most similar to the book as indicated with blue dotted-line. Thus, we can predict that User 1 will like the book. Even if User 1 has never rated the books previously, a CF system can still give a book recommendation. Though this capability can solve the problem of new users, particularly in a new domain, it is still difficult to make a prediction if there is no overlap between the book and movie ratings. Another limitation of cross-domain CF recommendations is that the item-user matrix is usually very large, and it can be much larger if it contains the users and items from multiple domains. This would result in extended processing time and heavy resource loading.

Items	User 1	User 2	User 3	User 4
Pirates of the Caribbean (movie)	like		like	dislike
1984 (book)	?	dislike	like	dislike
Father and Son (song)	dislike	like	dislike	like

Table 1. An example of a CF system that covers three domains: movies, books and songs

2.4. Content-Based Recommender Systems (CB)

In the content-based approach, a model for each user is developed based on the patterns and similarities in the content of previously rated items. After a user u_n gives a rating R_k on an item I_k, a training dataset TR_n that consists of pairs (I_k, R_k) is developed. Using this dataset, supervised machine learning algorithms are applied to create the user model which is used to predict the user's preferred items. The model can assess whether or not the user will be interested in an item that is not yet rated. Thus, the user profile has structural information about the user's preferences, and the system matches the profile and the items' descriptions in order to make recommendations [13]. In the following sections, we present: the way of representing items, the development of users' profiles using the adopted algorithm in this research: support vector machines, the advantages and disadvantages of CB and the process of making cross-domain recommendations in CB.

2.4.1. Item Representation in Content-Based RSs

The content depends on the item; for example, the features of a movie is its genre, actors, director, etc. [13]. It can be structured or unstructured; i.e. text. If the content is unstructured text that has no fixed features such as actors and directors etc., a preprocessing task that involves some complicated natural language processing needs to take place. The text needs to be tokenized then stemmed which means after the text is converted to separate words, the root of the word is found. Thus, words with different lexical class and similar meaning are converted to the same term. For example, "compute" is the root word for "computation," "computer" "computes" and "computers." [19]. After preprocessing the textual data, the term-frequency matrix, which represents items, is created. Term frequency $freq(d; t)$ means the number of times a term t occurs in a document d versus the number of all terms' occurrences in d [20].

2.4.2. User Profiles in Content-Based RSs

A user model has information about the user interests. The process of developing it is considered as a text classification task where the previously rated items are the training examples and their ratings are the target class. Machine learning algorithms are used to develop a function that learns from the previous behavior of the users to predict their next favored items. The algorithms may predict the numeric rating of the user; e.g., 5 stars or only predict if the user will like or dislike the item [19].

2.4.2.1. Support Vector Machines Algorithm (SVM)

It is a classification algorithm which in a linearly separable space discovers the maximum hyperplane that separates two target classes; e.g., like/dislike. Let L be the training set in the form of $(x_i; y_i)$. Each instance x_i contains D dimensions and is associated with a target class y_i= 1 or -1 where:

$i = 1 \dots L;\ x \in R^D;\ y_i \in (-1; 1)$. In this case an assumption is made that data is linearly separable so that a line, which is called hyperplane, can be drawn which is described by $w.x+b=0$ where:

- w is normal vector to the hyperplane [21].

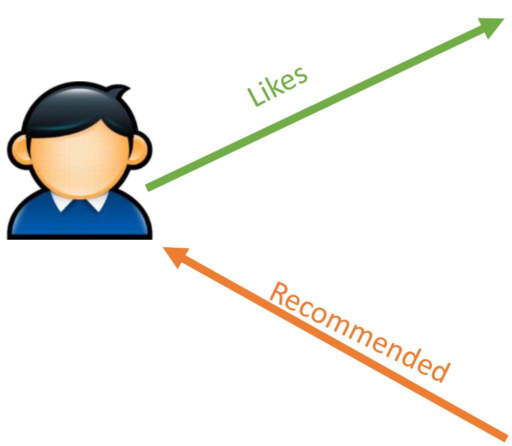
- $\frac{b}{w}$ is the distance between the hyperplane and the origin. The closest instances to separating hyperplane are called Support Vectors [21].

2.4.3. Advantages and Disadvantages

The CB system only focuses on one user's ratings to find patterns among them, and does not need the ratings of other users [13]. By not relying on users' ratings, it avoids having the gray sheep and shilling attack problems. In addition, the reasons why a CB system makes recommendations are known, since it makes suggestions based on item attributes. Thus, the system can justify why it recommended an item. For example, it could suggest an action movie with a specific actor to a user because the user tends to favor action movies with that actor. Moreover, unlike CF, content-based systems do not suffer from the new item problem, because as soon as an item is introduced, the system finds users who favored items with similar content, and recommends the new item to them [1] [13]. Nonetheless, CB does face some challenges. First, knowledge about items' content is required, and an inadequate item description could result in poor recommendations. Another issue is overspecialization; that is, when the system only recommends items with similar content to previously favored items. In addition, CB has the same new user problem as CF, in that a CB system can only build a representative model for a user when they have rated enough items. Otherwise, it has difficulty making quality recommendations [13].

2.4.4. Content-Based Cross-Domain Recommendations

In content-based recommenders, cross-domain recommendations can be based on direct relationships between domains; for example as shown in Figure 1, a RS can recommend a book to users because they have liked movies from the same genre. However, this method of recommendation is only applicable for similar domains [3].



Content of a movie

Genre	**Director**	**Cast**	**Year of production**
Fantasy	Catherine Hardwicke	Robert Pattinson	2008

Content of a book

Genre	**Author**	**Year of publication**
Fantasy	Stephenie Meyer	2011

Figure 1. Cross-domain recommendations in content-based recommenders

2.5. Other Types of Recommender Systems

In this section, four kinds of recommender systems are discussed: demographic, knowledge-based, social and hybrid recommender systems.

2.5.1. Demographic Recommender Systems

In this kind of RSs, personal information of users such as age, gender, country, education …etc. is exploited to build a user profile. The information might be collected explicitly by asking users to give information about themselves in the registration form or implicitly by tracking the users' activities. The recommendation process is done by correlating specific items to users' demographic classes. Demographics can be used to create stereotypes of users who like a particular item. For instance, the preferences of young adults are not similar to seniors. The advantage of demographic RSs is that, unlike CF, they do not rely on the ratings of users. Therefore, it does not suffer from the new user problem. Furthermore, this recommendation method does not depend on the domain of items as no information about items' content is required. However, privacy problems might be raised when collecting and using the personal information of users. In addition, the recommendation process is very simple to make efficient personalized system [22]. For example, not all American users who are older than 30 like the same book.

2.5.2. Knowledge-Based Recommender Systems

Knowledge-based RSs learn the users' needs or preferences and then ask its knowledge base for products that will satisfy these needs. For example, a user can choose what features he/she is looking for in a mobile phone then the RS look up the knowledge base and recommends the phones that meets the needs of the user. A Knowledge-based RS has many benefits. Its recommendation strategy does not need ratings from users and thus does not suffer from the new user problem. The system only takes into consideration the current preferences of users, thereby its recommendations are up-to-date. In addition, some items are suitable to be recommended by a knowledge-based RS such as cars, mobile phones…etc. That is because users look for specific features when buying these products. However, the system has many drawbacks. These systems require knowledge engineering which is a process of acquiring knowledge and representing it in a system that can reason and solve the issues based on given rules [14] [23]. This needs an understanding of the area of the recommended items especially to identify the features of a product that are significant and distinguishable [24] [25].

2.5.3. Social Recommender Systems

In the past few years, many social media websites arose which provide a place for individuals to connect with their friends and fellows. The rapid growth of Facebook, Twitter and other social media websites encouraged researchers to start considering them in relating to recommender systems in 1997. A social recommender is a systems that include the social media relations in its recommendation process. Users of social media can be members of a service, have friends and fellows or deal with trustworthy users. These relations are exploited to develop a social recommender system. An online friends would have more common interests with each other's than with users in the rating matrix. This type of recommendation consider two users as similar if they have a social relation. Let R be the rating of a user and $T \in R^{n*n}$ where T_{ij} is a social relation between two users u_i and u_j. If $T_{ij} = 1$, there is a relation between users and thus they are considered similar while if T_{ij} equals zero then no relation between them [26].

Social RSs require two components: users' ratings and their online social information. Similarly to CF, social-based RSs are categorized into: model-based and memory-based. The former uses CF models especially matrix factorization. The latter makes use of a memory-based CF (mostly user-based). To find similar users, most social-based RSs measure the trust between them and aggregate their ratings to predict the target user's ratings. One trust measure is MoleTrust which converts the trust network into a directed acyclic graph and then makes random walks on the graph to compute the trust values. Based on the ratings of users with the maximum-depth (a defined threshold), the estimated ratings of the target user are made [26].

2.5.4. Hybrid Recommender Systems

At least two RSs are joint in hybrid recommenders in order to enhance the performance or to overcome issues in one of the combined recommendations methods. For example, CF is usually associated with other recommendation methods such as social RS to avoid the new-user and new-item problems. There are seven techniques to develop a hybrid RS: Weighted: based on the votes made by the different recommendation techniques, the recommendations to the user are made; Switching: the hybrid RS changes the recommendation methodology after considering the present situation; Mixed: the user receives many recommendations made by different RSs; Feature combination: one algorithm makes use of features from different knowledge sources; Cascade: one recommendations technique receives the recommendations made by another RS and improves them; Feature augmentation: the features generated by one RS are used in the other RS; Meta-level: a RS is applied to produce a model which then used in another RS [14].

2.6. Personality in Recommender Systems

To solve many issues with the current recommender systems such as cold-start, and to exploit available features that might enhance systems' performance, researchers are investigating different fields, including psychology. In the following sections, the definitions of personality and the Big Five personality traits are given in addition to how were the Big Five personality traits developed and what methods are used to measure them.

2.6.1. Personality vs. Emotion

Personality forms in young adults, and remains essentially as it is for 45 years. It is not entirely static, but generally stable, and, thus, it is predictable [27]. A personality is "related to human thinking patterns, emotions and behaviors, together with psychological mechanisms behind those patterns." [28]. In contrast, emotions are temporary and continually changing. They are useful when RSs need to analyze a user's feelings, in order to provide recommendations that suit them in that moment. However, using personality is more appropriate when RSs need to predict products the user likes in general [27]. Consequently, in our work we exploit personality rather than emotions. Psychologists use different approaches when describing personality, and there are at least 18 theories to define it (e.g., the personality traits theory, used mostly in Affective Computing). Experts use the personality traits theory to find and measure the psychological differences among people. Computer systems can easily deal with personality when it is described with traits [29] [30].

2.6.2. What Are The Big Five Personality Traits (BF)?

It is a five-dimensional model that defines personalities on a broad level, and the dimensions represent the main differences among personalities [31]. According to [32], the BF personality traits represent "the relatively enduring pattern of recurrent interpersonal situations that characterize a human life." The BF traits, which also called dimensions or factors, are Extraversion (E), Agreeableness (A), Conscientiousness (C), Neuroticism (N) and Openness (O).

Each factor addresses many different personality characteristics as shown in Table 2 [9], and thus can provide adequate descriptions and facilitate making predictions [31].

Though these five factors describe the most significant personality traits, using them to categorize all human personality features means there are many different features under each factor, and this results in a significant loss of details. Thus, the Big Five dimensions can only describe personalities without detailed explanations. However, researchers can measure the Big Five factors and six subordinate traits, known as facets, associated with each BF factor by using a hierarchical model, such as the Revised NEO Personality Inventory (NEO PI-R) [31]. Table 2 [9] shows six facets that correlate positively with each BF dimension, and negatively with the opposite of the BF dimension. By using the NEO PI-R questionnaire or another method to obtain the BF traits as in section 2.6.4, five BF scores are given to the individuals. The higher a person's score in one dimension, the more applicable are its related facets. For example, if a user has a high score in the first dimension (Extraversion), it means they are sociable, forceful, energetic, adventurous, enthusiastic and outgoing, while a lower score indicates they are more introverted [9].

2.6.3. History of the Big Five Traits

The Big Five traits are not based on psychological theory, but on how personalities are described in natural languages. The first work in this area was done in 1936 by Allport and Odbert, who collected about 4,500 words that describe personality characteristics from English dictionaries. Statistical methods were used by Cattell in 1943 to group synonyms of Allport and Odbert's words into a shorter list, and this was the basis for Cattell's 16PF questionnaire, which consists of 16 personality factors. His work inspired other researchers, and led to the introduction of Big Five [33]. In 1961, Tupes and Christal found five factors that recur when they observed and analyzed Cattell's factors, and their work was replicated by Norman (1963), Borgatta (1964) and Digman and Takemoto-chock (1981). In the 1980s, many researchers agreed that these five recurring factors are essential to describe personalities; thus they were dubbed the Big Five by Goldberg in 1981 [9]. Later in 1999, Goldberg announced the International Personality Item Pool (IPIP) that can obtain the personality traits of individuals [34].

Big Five Dimensions	Facet (and correlated trait adjective)
E Extraversion vs. introversion	Gregariousness (sociable) Assertiveness (forceful) Activity (energetic) Excitement-seeking (adventurous) Positive emotions (enthusiastic) Warmth (outgoing)
A Agreeableness vs. antagonism	Trust (forgiving) Straightforwardness (not demanding) Altruism (warm) Compliance (not stubborn) Modesty (not show-off) Tender-mindedness (sympathetic)
C Conscientiousness vs. lack of direction	Competence (efficient) Order (organized) Dutifulness (not careless) Achievement striving (thorough) Self-discipline (not lazy) Deliberation (not impulsive)
N Neuroticism vs. emotional stability	Anxiety (tense) Angry hostility (irritable) Depression (not contented) Self-consciousness (shy) Impulsiveness (moody) Vulnerability (not self-confident)
O Openness vs. closeness to experience	Ideas (curious) Fantasy (imaginative) Aesthetics (artistic) Actions (wide interests) Feelings (excitable) Values (unconventional)

Table 2. The NEO PI-R facets of the Big Five Personality Traits

2.6.4. The Acquisition of Personality Traits

Researchers have tried to learn personality traits in four ways; through stories, keyboard, text, and questionnaires. To create a story-based personality learner, Dennis (2012) [35] took all the statements related to each BF dimension in a NEO-IPIP 20 questionnaire, which is a short version of IPIP [36], and created a story from them. For example, all the items in NEO-IPIP 20 under the Neuroticism dimension create the story: "Often feel blue. Dislike myself. Am often down in the dumps. Have frequent mood swings, etc." Test-takers were then asked to give a score, from 1 to 5, of the degree to which the story applied to the user, and it is found that some stories did indeed correlate with personality traits [30] [35]. Another method to obtain BF traits is keyword-based acquisition. In order to learn a person's typing features, the time taken while pressing keystrokes and between every two keystrokes is analyzed [37]. A study conducted in [38] on 20 test-takers whose mouse clicks and keystrokes were registered by software for eight days. They were asked later to fill IPIP-NEO 120-item questionnaire and their Big Five traits were calculated. Pearson correlation was applied between the users' usage of mouse and keyboard and their BF traits and facets, and a significant correlation was found. The study concluded that personality traits and their facets might be inferred by analyzing how people use the keyboard and mouse [38]. As well, users' generated textual information can be analyzed to recognize the Big Five personality traits. In an experiment in [29] the spoken and written language was given by participants in addition to their BF personality traits as reported by them and by observers. Machine learning models were trained to predict each BF value based on the participant's textual information. It is found that Extraversion and Openness are easily recognized from conversations and written text respectively. In general, the study shows that the recognition accuracy of all the BF traits based on conversational and textual language is at least 61% and 56% respectively [29]. Researchers are also investigating different fields, such as Kinect-based acquisition of personality information. In [39], a set of 108 videos of 12 mental states was analyzed. It is found that hand-over-face gestures are useful to recognize the mental states of users. However, the most widely used method to obtain BF traits is questionnaires. They can have different ranges of questions, and the more questions there are, the more fine-grained traits can be obtained [30].

2.6.4.1. Big Five Traits Questionnaires

There are 13 Big Five-based questionnaires used to measure personality differences, and the questionnaires that measure the facets are usually long and time-consuming [27]. Two of the best known questionnaires, NEO Personality Inventory Revised (NEO-PI-R) and Big Five Inventory (BFI) [15], are addressed here.

By statistically analyzing Cattell's 16PF, McCrae and Costa (1985) constructed the NEO Personality Inventory, which is named for the N, E and O factors. Later, the other two factors were included in a new version of the inventory, known as the NEO Personality Inventory Revised (NEO-PI-R). This version consists of 240 questions that measure the Big Five factors and the six facets that fall under each factor, as shown previously in Table 2. However, with so many questions NEO-PI-R takes a lot of time to complete [33] and for this reason we chose to use another questionnaire for our work.

Though many researchers agreed on the initial five main factors, their five factors were given different names, and each addressed different facets. Thus, there was a need to identify the common traits each different study had under each factor, and research was conducted to achieve this. The result was a list of the common traits, or the "core" of the Big Five, as identified in

Table 3 [33]; the Big Five Inventory (BFI) was developed based on these core traits. BFI is a 44 item questionnaire with eight to ten questions for each BF dimension, as shown in Appendix A, and it is considered a short version of NEO-PI-R [40]. Although NEO-PI-R is the most accurate and widely used inventory in the literature, BFI is preferred when the test-takers' time is limited. In an experiment in [33], the test-takers needed only five minutes to complete the BFI questionnaire. In addition, BFI's items are not just adjectives (as in IPIP) or long sentences (as in NEO-PI-R), but actual phrases that are more understandable [33].

Extraversion		Agreeableness		Conscientiousness		Neuroticism		Oppenness/Intellect	
Low	High	Low	High	Low	High	Low	High	Low	High
-.83 Quiet	.85 Talkative	-.52 Fault-finding	.87 Sympathetic	-.58 Careless	.80 Organized	-.39 Stable*	.73 Tense	-.74 Commonplace	.76 Wide interests
-.80 Reserved	.83 Assertive	-.48 Cold	.85 Kind	-.53 Disorderly	.80 Thorough	-.35 Calm*	.72 Anxious	-.73 Narrow interests	.76 Imaginative
-.75 Shy	.82 Active	-.45 Unfriendly	.85 Appreciative	-.50 Frivolous	.78 Planful	-.21 Contented*	.72 Nervous	-.67 Simple	.72 Intelligent
-.71 Silent	.82 Energetic	-.45 Quarrelsome	.84 Affectionate	-.49 Irresponsible	.78 Efficient	.14 Unemotional*	.71 Moody	-.55 Shallow	.73 Original
-.67 Withdrawn	.82 Outgoing	-.45 Hard-hearted	.84 Soft-hearted	-.40 Slipshot	.73 Responsible		.71 Worrying	-.47 Unintelligent	.68 Insightful
-.66 Retiring	.80 Outspoken	-.38 Unkind	.82 Warm	-.39 Undependable	.72 Reliable		.68 Touchy		.64 Curious
	.79 Dominant	-.33 Cruel	.81 Generous	-.37 Forgetful	.70 Dependable		.64 Fearful		.59 Sophisticated
	.73 Forceful	-.31 Stern*	.78 Trusting		.68 Conscientious		.63 High-strung		.59 Artistic
	.73 Enthusiastic	-.28 Thankless	.77 Helpful		.66 Precise		.63 Self-pitying		.59 Clever
	.68 Show-off	-.24 Stingy*	.77 Forgiving		.66 Practical		.60 Temperamental		.58 Inventive
	.68 Sociable		.74 Pleasant		.65 Deliberate		.59 Unstable		.56 Sharp-witted
	.64 Spunky		.73 Good-natured		.46 Painstaking		.58 Self-punishing		.55 Ingenious
	.64 Adventurous		.73 Friendly		.26 Cautious*		.54 Despondent		.45 Witty*
	.62 Noisy		.72 Cooperative				.51 Emotional		.45 Resourceful*
	.58 Bossy		.67 Gentle						.37 Wise
			.66 Unselfish						.33 Logical*
			.56 Praising						.29 Civilized*
			.51 Sensitive						.22 Foresighted*
									.21 Polished*
									.20 Dignified*

Table 3. A list of central trait adjectives and factor loadings consensually selected by 10 psychologists

3. Related Work

In this chapter, an overview of the recommender systems that have adopted personality information in general and the Big Five personality traits in particular is presented. Furthermore, different approaches of recommender systems that have addressed the cross-domain issue are discussed.

3.1. Personality and Recommender Systems

A study by Hu and Pu [41] compared recommendations from two techniques: personality-based as implemented in Whattorent.com [42], and rating-based as in movielens.org [43]. MovieLens, a website provided by GroupLens Research, is an example of a traditional rating-based collaborative filtering system. As its name suggest, it recommend movies to users. As new users start consuming MovieLens, they are asked to give ratings for fifteen movies that they have already watched. Based on the given ratings by a user, her/his profile is developed. A user profile in a rating-based CF is a vector that has the ratings for items as given by the user. MovieLens then compare the profile of users to find the most similar ones which are called "neighbors". The system uses the ratings of these neighbors to predict the possible ratings of the target user. The movies recommendations are then presented to the user from in descending list [27] [43].

The personality-based RS implemented in Whattorent.com [42] relies on the LaBarrie Theory to make movie recommendations. According to this theory, "A movie viewer emotionally interacts with a film in the same manner that they interact with other human beings." All movies in the database are considered as people, and their emotional effect is analyzed. Whattorent users answer a 20-item questionnaire that interprets their reactions to real situations, and analyzes their average psychological condition. The users also indicate their current mood, and which movie experience they are looking for. This user information is then matched with movies in the database, and recommendations are provided. According to the website, users typically consider themselves different from what they really are, so the quiz includes questions about regular daily events because they can lead to more accurate personality definition. Users of Whattorent's personality-based RS were satisfied with 80.27% of the recommended items [42]. Unlike the Whattorent.com approach, our system does not require knowledge engineering of the effect of each item in order to match it with users' personalities. In addition, the applied personality-based theory is different, as [42] does not use the Big Five traits. Hu and Pu [41] evaluation schema of whattorent.com and movielens.org had three factors, namely perceived accuracy, user effort and user loyalty. Thirty test-takers used both systems, and the personality-based technique was rated highest because it required less time and effort than the rating-based technique. As well, users said they would probably visit websites with personality-based recommendations more frequently than other websites. Therefore, it was concluded that personality-based recommenders have the potential to be successful systems.

Hu and Pu investigated personality-based RSs further in [44]. They based this work on Rentfrow's findings [7], which determined the correlation between people's preferences in the music domain and the BF personality traits, as shown in Table 4. The system in [44] makes recommendations after matching a user's personality with the positively correlated musical preferences groups. Their music dataset had 1,581 songs, and eighty users completed a Ten-Item Personality Inventory (TIPI) questionnaire for themselves and their friends. Based on their BF personality traits learned from the questionnaire, the system recommended 20 songs for them and their friends, and asked them to rate them. The results demonstrated that songs sent to friends as gifts were liked by the recipients, and that the test-takers enjoyed taking the test. Hu and Pu used a knowledge-based recommending system, with static facts about how personalities correlate with music genres. They have not done any knowledge engineering, yet applied Rentfrow's findings in [7]. However, our approach does not require knowledge engineering, and is applicable not only to music, but all types of products. For example, in our approach, the user with high Conscientiousness would not be necessarily recommended with country, religious and pop music as Table 4 instructs. That is because in our system, we do not rely on predefined facts as in Rentfrow's findings to make recommendations. Rather, the system is flexible and a jazz, pop or hip-hop music or even other types of items such as books can be recommended to a user as long as these items have been preferred by users with similar personalities as the items the target user has liked.

Musical preferences	Example	Positive correlation with Big Five traits	Negative correlation with Big Five traits
reflective and complex	blues, jazz, classical and folk	Openness	-
intense and rebellious	rock, alternative, and heavy metal music	Openness	-
upbeat and conventional	country, religious, and pop music	Extraversion, Agreeableness and Conscientiousness,	Openness
energetic and rhythmic	rap/hip-hop, soul/funk, and electronic/dance music	Extraversion and Agreeableness	-

Table 4. The correlations found by Rentfrow between musical preferences and personality traits

Other research [6] studied the correlation between 16 genres of TV shows, movies, songs and books, and the BF personality traits. Examples of genres in TV shows domain are: animation, cooking show, documentary, talk show; in the movies domain: comedy, action, drama, and science fiction; in the books domain: comic, crime, drama, educational; and in the music domain: blues, classical, country, hip hop…etc. The researchers in [6] have used the IPIP questionnaire to obtain the BF traits of users, then averaged the BF values of the users who like each genre and the result is called the stereotypical personality of that genre. For instance, after finding the average score of the BF personality of 722 users who like documentary shows, it is found that the stereotypical personality that prefers documentary genre as shown in Table 5 has high C, E, A, O and low N. In our work, we examined two methods to develop item profiles. In one of them, the same technique used for constructing stereotypical personality in [6] is adopted in our RSs. More details are given in section 4.1.2. Their findings were not used to make recommendations, but, as with Rentfrow's work [7], they led us to believe that it is possible to find a relationship between each item (e.g., the book '1984') and personality traits. However, our approach considers the personalities of users who like a products without taking into account any information related to different genres.

TV Genre	O	C	E	A	N	#users
documentary	4.12	3.45	3.37	3.53	2.85	722
news	3.97	3.58	3.58	3.54	2.74	676
reality show	3.76	3.56	3.61	3.58	2.75	808
talk show	3.81	3.58	3.58	3.68	2.67	61

Table 5. Stereotypical personality of users in 4 TV genres, the highest score is 5

In another work, Hu and Pu [45] applied a user-based collaborative filtering system that utilizes the BF personality traits of users, employing the same dataset as in [44], and calculated the similarity between users' Big Five personality traits with the Pearson correlation. They tested three techniques: pure personality-based CF, the linear hybrid approach (personality-based and rating-based), and the cascade hybrid approach. The results show that the prediction accuracy of the cascade hybrid approach is higher than the other methods, and both the pure personality-based and linear hybrid approaches were more accurate than the rating-based CF. However, when compared to the other approaches, the cascade hybrid system took the most time to make recommendations, and the pure personality-based CF took the least. In addition, enriching the user-based CF with user personality information helped address the user cold start situation. That is because the systems forms neighbours for new users, who have not rated any item yet, based only on the similarity of their BF traits with other active users.

A similar system, implemented by Tkalcic in [46], used IPIP inventory (50 questions) to obtain users' BF values. Each user was modeled with the five BF personality dimensions, and collaborative filtering based on the similarity between the BF values was developed. This approach was compared with the traditional CF system, and its performance was statistically similar or better. In addition, the proposed system was not as computationally expensive as the rating-based CF because personality is static, and the system forms neighbors that do not change with time as they do with the rating-based CF. However, the system still requires users to explicitly rate items. Both [45] and [46] are similar to our approach, as they are collaborative filtering systems that use the BF traits. Thus, we also compared the recommendation accuracy to the traditional CF when evaluating the system. However, our proposed system differs from [45] and [46] in the following ways:

- It is not user-based CF but an item-based CF that defines each item using the BF traits. Our system does not compare the users' personality profiles to each other. Rather, it calculates the similarity between items personality profiles. Experiments in [18] show that the results of item-based CF are far superior to those of user-based CF.

- It adopts a new technique for developing probability-based items personality profiles as described in section 4.1.2.
- It addresses the cross-domain recommendation problem.

One of the pioneer works in personality-based recommender systems was proposed by Nunes [27]. She surveyed the different approaches to describe personality such as Cognitive, Behavioral…etc., and determined that personality traits are best for modelling personalities in computers. The traits were extracted at two levels: coarse-grain and fine-grain. The TIPI questionnaire (10 questions) was used in the former, and the NEO-IPIP inventory (300 questions) in the latter. Nunes also proposed two methods of personality representation in computers: User Psychological Profiles, which are used to include an individual's "Internal Identity", that is, how the individual describes their own personality; and User Psychological Reputation, which is an individual's "Social Identity", as described by their friends. In one experiment, ten users completed three NEO-IPIP Inventories (300 questions each) to describe the personality of their perfect president, and two current candidates. A reputation profile was generated for each candidate, by summarizing their personality as described by the users. Then a comparison between candidates' reputation profiles and each user's perfect president was done, with the target class as the user real vote; the result was 80% accurate. In another experiment, 280 students completed TIPI, and then formed groups based on their personal choices. Later, a recommender system was applied, using students' choices as the target class. The system achieved 100% prediction accuracy in 4 of 19 groups. Unlike our work, Nunes's is designed to be a social matching RS that suggest friends or colleagues and it is not useful when recommending products.

3.2. Cross-Domain Recommendations

Winoto and Tang are two of the first researchers to investigate cross-domain RSs [2]. They found three potential issues when a cross-domain RS is developed: finding the relationships between the user's preferences in different domains, developing a method to transfer knowledge about the user from one domain to predict their preferences in another domain, and proposing an evaluation methodology that is suitable for cross-domain RSs [2].

In [3], cross-domain recommenders are categorized into three categories. The first is to merge user preferences from multiple domains into one user profile. This was implemented by [47], which integrated users' tag-based profiles in Flicker and Delicious, social media sites, in order to increase their interest. The second method is to use domain related features to establish relationships between domains. An example of this was done in [48], which connected multiple domains with knowledge-based rules that were used to provide recommendations. Their work depends on establishing rules between the features (e.g., genre) in two domains. For instance, one rule that provides a bridge between the domains of movies and books is "People who like movies in romance drama group also like books in dramatic poetry". The third approach is to use transfer learning, as implemented in collaborative filtering in [49]. Here, the user-item matrix is clustered and rating patterns are determined, and these are used later to create relationships between domains. An illustration of the process of transfer learning between movies and books domains is shown in Figure 2. After clustering the original user-item matrix, it is found that user clusters {2, 3}, {1, 5}, {4, 6} and item clusters {a, e}, {b, f} from the movies domain can be matched respectively with user clusters {1, 4, 5}, {2, 6}, {3, 7} and item clusters {c, d}, {a, b, e} in the books domain. The rating pattern (codebook) from the movies domain is transferred to be expanded in the sparse books matrix as shown in Figure 3.

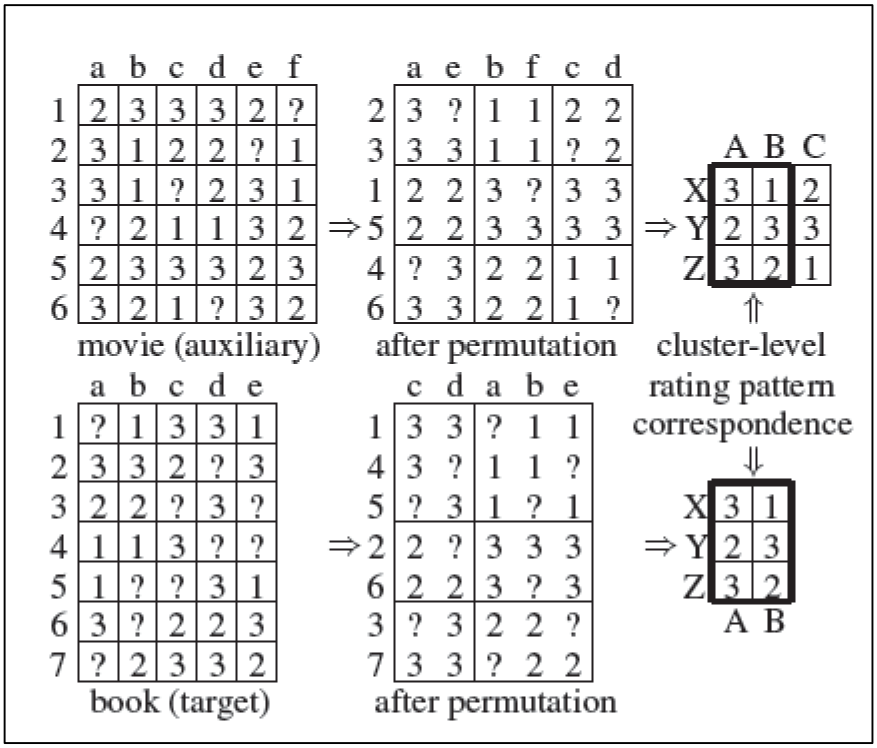


Figure 2. The correspondence of two rating patterns between the domains of movies and books

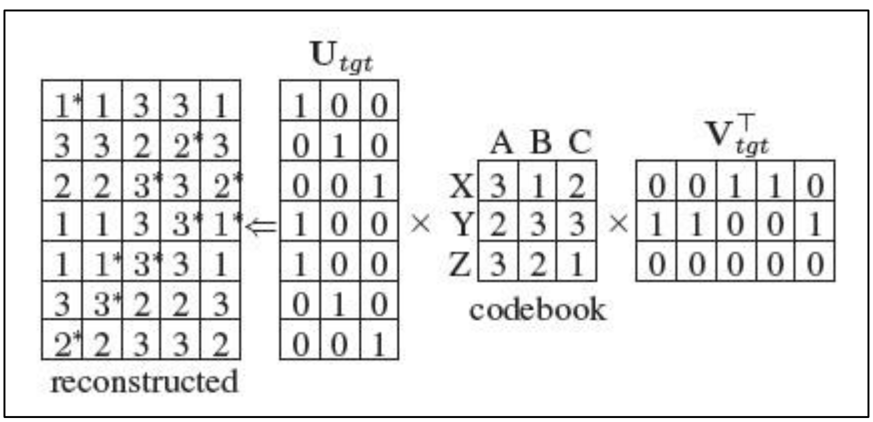


Figure 3. The reconstruction of books rating matrix.
Values with '' are new and values without '*' are the same as the original book matrix*

Our system would not be categorized under any of these approaches, as it does not need to combine user profiles of domains, construct rules and find relationships, or transfer learning between domains. The keys to cross-domain recommendation in our system is that it focuses on the users' personalities not on the features of items. It represents products from different domains by using specific numbers of attributes that reflect the personality of the users who like the products. In addition, to the best of our knowledge, our research is the first that exploits BF personality traits in making cross-domain recommendations.

One research [50] exploits the availability of users' cross-domain preferences on social networks especially Facebook to make cross-domain recommendations. After extracting the mentioned preferences from the generated text of 95 users and their friends, their Facebook preferences were added to a CF system to enrich the ratings matrix and enhance the performance. It is found that some features played a role in enhancing cross-domain recommendations including the number of common users who have preferences in the two domains and the number of items that the user has mentioned. Our proposed approach does not require information extraction. Moreover, the BF personality traits acquisition would takes place once (at the registration) whereas extracting mentions would be repeated to update the user's profile. In addition, our work focuses on users' personalities, and thus finds deeper associations between users and items.

4. The Method

Since the introduction of RSs in the nineties, they have been an active research area [1] [3]. However, cross-domain recommendation has not been explored to the same extent as single-domain [2] [3]. Most current RSs lack the framework from which cross-domain recommendations for any domain can be made. They are domain-specific, and only work for some types of products after they find correlations among the types. For example, an RS may be able to give recommendations across book and movie domains after it finds relationships between the domains, but it cannot do this for all other kinds of products.

What is required is a base from which recommendations from all the domains can be developed. The items in such a system should be represented in the same way so that the similarity between the items from different domains can be found. Representing the items using their content results in having domain-specific system whereas modeling them by users' ratings lead to having a very large ratings matrix which is hard to process as discussed in section 2.3.3 and 2.4.4. As the user is the center of the recommendation process, concentrating on their information could be the key to a successful cross-domain RS. The user is the common factor of every product they purchase. Therefore, a system that incorporates users' information would likely be very useful for cross-domain recommendation.

Users' psychological information might be highly important to achieving cross-domain recommendation. The personalities of users have been successfully applied in RSs to increase recommendations accuracy [44] [46], as well as to address sparsity and new user problems [45] as explained in section 3.1. However, we are not aware of any research in the RSs area that used BF traits to promote multiple-domain recommendations. We believe that taking personality information into account when generating recommendations could reveal meaningful associations between users and items, and lead to more effective cross-domain RSs. Our belief is triggered by [6] and [7] which, as discussed in section 3.1, found a relationship between peoples' preferred genres and their psychological makeup.

In the proposed system, we capture the personality trait(s) of the fans of an item, and use them in RSs. We suggest a new framework for cross-domain recommendations, in which all products are represented by a specific number of features that reflect the personality of the products' fans. These features formulate the Item personality profiles (IPPs) which are used for item-based collaborative filtering.

4.1. Description of Collaborative Filtering Based on Items' Personality Profiles (IPPs-based CF)

The proposed system follows the collaborative filtering recommendation methodology, but it differs from a traditional CF in the way items are represented. In our approach, after the personality of users is determined, either explicitly (e.g., via questionnaires) or implicitly (e.g., by analyzing a user's generated text), a personality profile is constructed for each user. Once an item is liked by a number of users, its item personality profile (IPP) is developed based on the personality profiles of those users. IPP is a vector consisting of 5 or 15 attributes that aggregate the personalities of users who like the item. The similarity between IPP vectors is then found, and users are recommended items that are similar to those they already prefer.

The first step in building an IPPs-based CF system, as shown in the system workflow (Figure 4), is to learn the personality of the users. In our implementation, users were asked to complete the BFI questionnaire (any other methods to acquire the users' BF personality traits can also be used). In the case of questionnaires, the user answers must be transformed into five scores that represent the BF values of a user. They are transformed by performing the calculations indicated by the developers of the adopted questionnaire. In the second step, a user's personality profile (their five BF values) is stored in a repository, and when required can be accessed by the IPP generator, which is responsible for the development of IPPs. Thirdly, users' feedback on items is obtained either implicitly (i.e. learned from users' behavior) or explicitly (i.e. by their ratings on the items). Only a user's positive feedback, as indicated by the green arrows in step 4, is sent to the IPP generator. That is because a product's IPP should only reflect the personalities of the fans of the product. In the fifth step, when an item receives positive feedback from users, the IPP generator retrieves the users' personality profiles from the repository, and builds the item's personality profile (IPP). Lastly, the IPPs are sent to the CF algorithm, which calculates the similarity between them and makes recommendations to the target user.

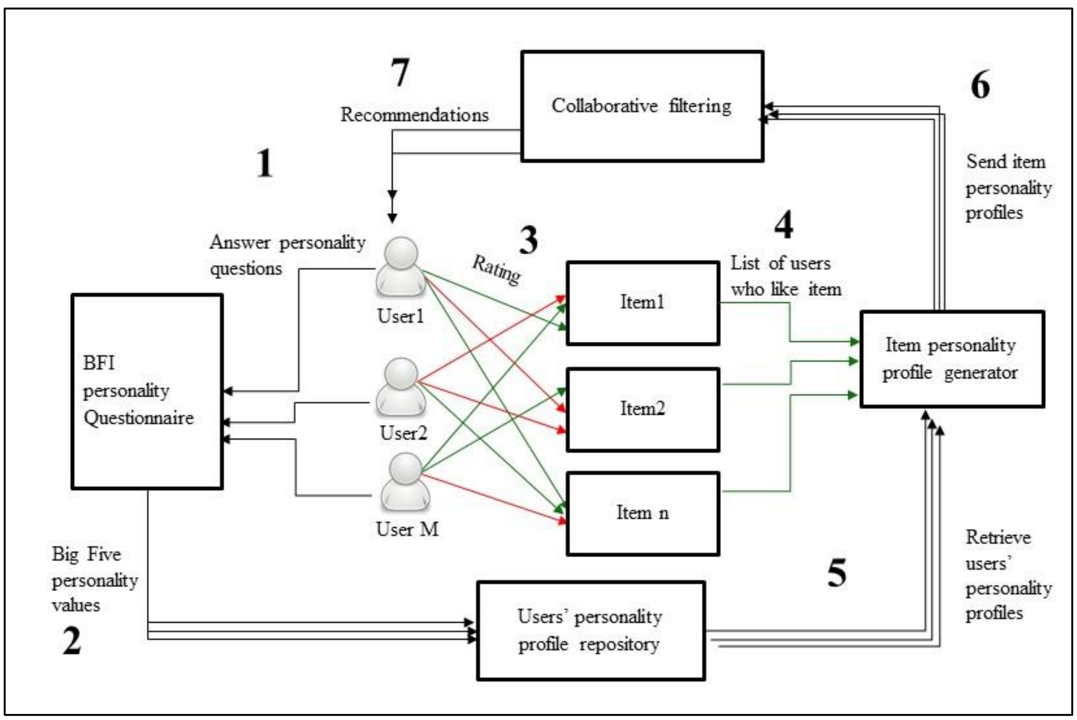


Figure 4. Workflow of the item-based collaborative filtering using the Big Five personality traits

The following sections explain the implementation of the three main steps in the IPP-based CF: (i) the development of the personality profiles of the users, (ii) the construction of the personality profiles of items, and (iii) the use of IPPs in CF recommendations.

4.1.1. The Development of the User Personality Profile

As discussed in section 2.6.4.1., we used the Big Five Inventory (BFI) which is provided in [51]. After receiving users' answers on the BFI, we transformed the 44 answers into 5 personality traits; this is considered the user personality profile. In the Big Five Inventory, eight questions are associated with each E and N, nine with each A and C, and 10 with O, and each answer is in the range of 1 to 5. To calculate the scores for the five dimensions we followed the instructions in BFI which is developed by Oliver P. John (1991) in [33], [52] and [53]:

1- Find the total score of all questions related to each BF dimension (E, N, A, C and O).

2- Calculate the mean by dividing the total score by the number of questions. We call this the user BF dimension mean.

3- Standardize the result by converting it to a Z-score, where:

> Z = (a user BF dimension mean – a given population BF dimension mean) / a given population BF dimension standard deviation (SD).

4- Convert the Z-score to a T-score, with a mean of 50 and a standard deviation of 10. The lowest T-score is 20 and the highest is 80.

Each BF trait has different number of questions, population BF dimension mean, population BF dimension standard deviation (SD) as shown in appendix A. For example, the following steps are applied in order to convert total score of all questions related to Extraversion into a T-score for any user.

Extraversion___. Total Score divided by 8 = ____ (X). X minus 3.2 = ____ (Y). Y divided by 0.8 = (Z) = ____. (Z * 10) + 50 = ____ (T)

In case the total score of all questions related to Extraversion for a user is 23, then the T-score or Extraversion value of the user is 45.94 as explained in the following steps:

Extraversion 23. Total Score divided by 8 = 2.88 (X). X minus 3.2 = -0.325 (Y). Y divided by 0.8 = (Z) = - 0.406. (Z * 10) + 50 = 45.94 (T)

The result is five T-scores which are considered the BF values. A BF value is considered 'low' if it is less than 45, 'high' if it is greater than 55, and 'Average' otherwise [27]. In Table 6, the BF values for one user and their related high, average or low degrees are presented. The five BF values of a user is her/his personality profile.

The form of the personality profile	Extraversion	Agreeableness	Conscientiousness	Neuroticism	Openness
User 5 personality profile (values)	47.50	53.33	58.89	71.88	38.57
User 5 personality profile (degrees)	Average	Average	High	High	low

Table 6. One user personality profile in two forms

4.1.2. The Development of Items' Personality Profiles (IPPs)

The assumption behind this work is that items are liked by users who have some common personality features. Thus, the personality profiles of only the users who like an item are used to create that item's personality profile. Since some researchers such as [6] and [7] found a correlation between users' preferences and their BF traits, we think there is an evidence that BF traits play a role in determining the positive ratings of users (their preferences). However, we cannot determine that products which users do not like correlate with their BF traits, thus we did not use negative ratings in developing IPPs. In addition, many e-commerce organizations learn the preferences of users as unary ratings because the absence of ratings is not considered as negative [54]. An example of explicit unary ratings is shown in Facebook which allows users to press the 'like' button to express their preferences while implicit unary ratings can be learned from the user behavior (e.g., purchased/view the product) [55]. The fact that our proposed system require only positive ratings to create IPPs makes it able to work not only on systems that provide the like and dislikes of users but also to work with systems that learn only the unary ratings.

We tested two methods to generate item personality profiles. In the first which we call the average-based IPPs, after isolating one item and eliminating users who dislike it, we found the arithmetic mean of the BF values of users who like the item. The result was five personality dimensions, and these were considered to be the item's personality profiles. This method of building IPPs was inspired by [6], which found the mean of users who like a genre. In Table 7, the average-based IPPs for three items are illustrated. However, calculating just the average BF personality of users only captures the central tendency, and neglects a variety of users' personalities.

Movie	Extraversion	Agreeableness	Conscientiousness	Neuroticism	Openness
Finding Neverland	47.50	51.88	51.23	50.68	53.91
Pirates of the Caribbean	48.61	49.99	49.37	51.92	52.12
127 Hours	48.66	51.55	50.84	51.85	52.64

Table 7. The average-based IPPs of three movies

In the second technique to produce item personality profiles called the proportion-based IPPs, we represented an item personality profile as a vector, consisting of 15 features. Every three consecutive features represent three degrees (low, average and high) of one BF dimension (e.g., extraversion). The value of one feature is the proportion of users who like an item in the degree of one BF dimension. The steps to develop the proportion-based IPPs are as follows:

- For each item, the records of users who disliked it were deleted, and the records of those who liked were retained.
- Then, the number of users with low, average and high values in each BF dimension were totaled. Table 8 illustrates this process for the movie ‘Insidious’.
- The proportion of users with low, average and high values in every BF dimension was determined.
- Table 9 shows the proportion of users for this movie. Three fans of the movie have a low value in the extraversion dimension, and the total number of users who liked it was 15. Thus, as seen in Table 9 the proportion is 3/15 = 0.20.
- The last form of the item personality profile is a vector consisting of the 15 features that reflect three degrees in each of the five BF dimensions. Table 10 shows the final form of the four different item personality profiles.

The degree of BF value	E	A	C	N	O
Low	3	5	6	1	4
Average	6	5	6	11	5
High	6	5	3	3	6
Total number of users who like the movie Insidious	15				

Table 8. The number of users who have low, average and high values in each BF dimension for the item ‘Insidious’

The degree of BF value	E	A	C	N	O
Low	0.2	0.33	0.4	0.07	0.27
Average	0.4	0.33	0.4	0.73	0.33
High	0.4	0.33	0.2	0.2	0.4

Table 9. The proportion of users who have low, average and high values in each BF dimension for the item 'Insidious'

Item	E - low	E - ave	E - high	A - low	A - ave	A - high	C - low	C - ave	C - high	N - low	N - ave	N - high	O - low	O - ave	O - high
127 Hours (movie)	0.26	0.52	0.22	0.15	0.56	0.30	0.15	0.56	0.30	0.19	0.48	0.33	0.26	0.22	0.52
Insidious (movie)	0.20	0.40	0.40	0.33	0.33	0.33	0.40	0.40	0.20	0.07	0.73	0.20	0.27	0.33	0.40
As You Think (Book)	0.19	0.52	0.29	0.14	0.67	0.19	0.24	0.43	0.33	0.19	0.48	0.33	0.19	0.19	0.62
The Alchemist (Book)	0.19	0.54	0.27	0.23	0.46	0.31	0.31	0.46	0.23	0.19	0.5	0.31	0.19	0.27	0.54

Table 10. The final form of the item personality profiles for four items

Of course, users with different personality traits, and sometimes even opposite personalities, could still like the same item. For example, an extrovert and introvert may like the same movie. However, in this methodology we only include their number as part of the total number of fans. The presence of a few users with a certain trait's degree (high, low or average) will not greatly affect the representation of an item. In contrast, the average-based IPPs is sensitive to few users with very high or low BF score because of the nature of arithmetic mean. In contrast to average-based IPPs, the proportion is not affected by a bias BF value (e.g., a very high value), and it allows for a representative profile that shows all the users' various personalities. In addition, in the proportion-based IPPs, there could be an item that has an equal number of raters under all the traits' degrees. However, that would mean that the item is preferred by different personalities, and it will share similar representation with other items in the system. In our experiments, we tried both of the methods and reported their results.

4.1.3. Item-based Collaborative Filtering Algorithm

After constructing the item personality profiles, we used them to create item-based CF recommendations. In our system, the algorithm calculates the similarity between the IPPs of items that the user have liked before and the IPPs vectors of other unrated items, and finds the k most similar items to recommend to the user. When implementing CF, we used the K-nearest neighbour algorithm and Euclidean distance [20], which were discussed in section 2.3.2.1, as implemented in WEKA, a machine learning tool [56].

5. Evaluation

Recommender systems are evaluated in three ways: offline, online and via user studies [4]. We conducted an offline experiment that assumed the user will act similar in real-time to how they acted before the data is collected. This is done by hiding part of the user's feedback in order to use it as test dataset, then instructing the RS to make predictions based on what it learned from the user's unhidden behavior. The result of the recommendations can be one of the four cases shown in Table 11. The advantage of offline experiments is that, unlike online experiments, they can compare algorithms without making wrong recommendations to real users, while their drawback is that they only focus on limited factors of ensuring users' satisfaction, including the accuracy [4].

True Positives The set of relevant items that are recommended (i.e. correct results).	**False Negatives** The set of relevant items that are not recommended (i.e., missing correct results).
False Positives The set of non-relevant items that are recommended (i.e. incorrect, unexpected results).	**True Negatives** The set of non-relevant items that are not recommended (i.e. correct absence of results).

Table 11. The confusion matrix. Relevant items are items liked by the user and non-relevant items are those disliked by the user.

5.1. Dataset

Testing the capability of the suggested approach to make recommendations across-domains requires a dataset consisting of users' preferences in multiple domains, in addition to their personality information. One available dataset, from the mypersonality project [57], has users BF traits and preferences in many domains, as shown in the users' Facebook accounts. However, these users did not provide negative ratings, so when the suggested RS makes recommendations we would not be able to determine if the user dislikes them. Thus, we had to collect the required data for the evaluation. Forty-six colleagues, friends and family members completed an online survey, which had three sections: the Big Five Inventory, the ratings for ten movies, and the ratings for five books. Before taking the test, we guaranteed the users would receive an email informing them of the values of their personality traits, and we assumed they answered the personality questions honestly, in order to get true BF values. The answers to the personality questions were automatically calculated in a Google spreadsheet[1], and converted to one of three categories: low, average or high. Then an email was immediately sent to the test taker with their BF values and description. Similar to [27], the BF value is considered 'low' if it is less than 45, 'high' if it is greater than 55 and 'average' otherwise.

The selected items were of diverse genres, so they would appeal to different personalities. Users rated most of the movies and books, and the movie ratings were in the range of 1 to 4 stars. As for the book ratings, we expected that not many users would read all five books. Thus, if users had not read a book, we asked them to indicate whether or not they would be interested in reading it. If a user had read a book and they liked it, or were interested when reading it, we classified it as 'like', and the opposite as 'dislike'.

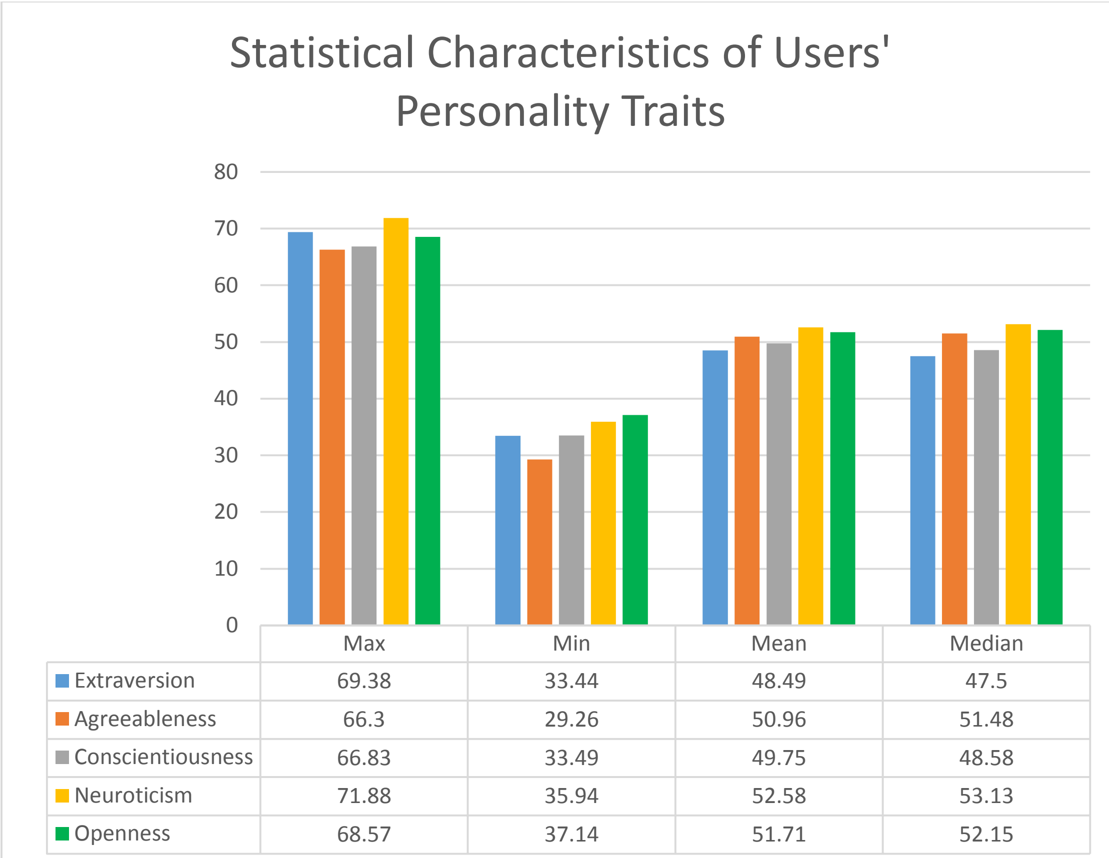
Since the questions were multiple choices, there were no noisy data; however, two users completed the survey twice. One informed us that she did it twice because the first time she did not understand the language of the survey; her first record was deleted. The second user had two identical records, so one of them was deleted.

After cleaning the data (the deletion of duplicates), the overall number of users was 46 consisting of 25 females, 19 males and two unknown. Other research has focused on the different

[1] http://www.google.ca/sheets/about/

preferences and personalities of the two genders [6], but we did not consider gender when developing the IPPs. Most users, 61%, were in their twenties, 17% were in their thirties, 14% were 40 years old or more, and only one was under 20. As mentioned previously, most psychologists believe that personality is almost stable for 45 years [27]. In addition, most test-takers are relatives from Arab countries, and six test-takers are south Asian. Researchers [58] have discovered a relationship between the BF personality traits and different cultures. We expect that any work that includes personality information could be affected by users' gender and cultural background. However, as far as we know, none of the recommender systems that collect personality information take that information into account. As well, if the user is a fan of an item, their personality is captured in our system because our goal is item personality profiles that include all fans' BF traits, regardless of their gender or demographics.

The central tendencies of users' Big Five personality traits are presented in Figure 5. As shown, the highest BF value is 71 (Neuroticism), and the lowest is 29 (Agreeableness). The mean BF value of all users ranges from 48 to 52, which means that they all fall into the average category. Additionally, the number of users who have low, average or high BF values are aggregated, as depicted in Figure 6. As shown, in the first four dimensions the majority of users have average BF values, but in Openness high BF values are dominant.



	Max	Min	Mean	Median
Extraversion	69.38	33.44	48.49	47.5
Agreeableness	66.3	29.26	50.96	51.48
Conscientiousness	66.83	33.49	49.75	48.58
Neuroticism	71.88	35.94	52.58	53.13
Openness	68.57	37.14	51.71	52.15

Figure 5. Maximum, minimum, mean and median of the BF values of all users

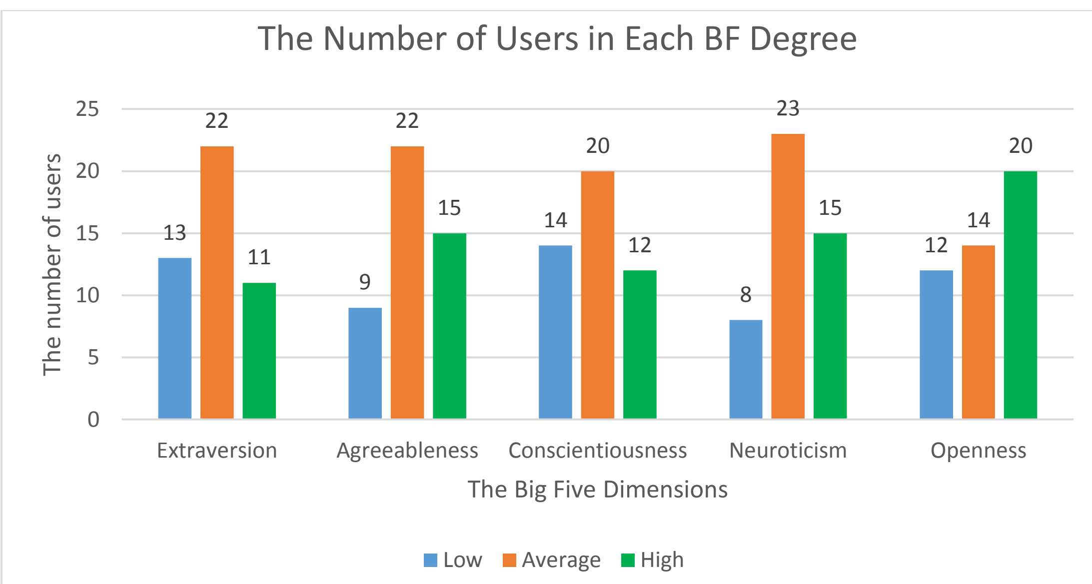


Figure 6. The number of users with Low, Average and High BF values

The overall number of users and items in the dataset is shown in Table 12. In order to indicate the percentage of missing ratings, we calculate the sparsity level similar to [59], as in Eq. 2. For example, to calculate the sparsity level of the entire dataset, we perform the following: Sparsity level= 1-(624/690). A value of 10% sparsity means that the user-item matrix is not highly sparse. It is clear that the dataset is not highly sparse because the number of items is less than the number of users. In reality, users rate a small subset of the overall number of items which results in a sparse dataset. However, in our experiment, users rated most of the items because the number of items is small.

Eq. 2 *Sparsity level = 1- (given ratings / (number of users* number of items))*

Users who provided a rating for only a few items, and those who had unary ratings of ‘like’ or ‘dislike’, were not used as test cases. After they were excluded, the final number of test cases that could be used as the target class in the movies domain was 36, and in the cross-domain (both movies and books) it was 22.

5.1. Evaluation Metrics

The ultimate goal of RSs, as perceived by many researchers, is to provide the customer with recommendations that are as accurate as possible. Thus, the accuracy of recommendations is the most important aspect when evaluating RSs [4]. The metrics to evaluate the accuracy were categorized by [4], into ratings predictions, usage predictions and rank accuracy metrics. The following subsections discuss the metrics of ratings predictions and usage predictions in addition to the statistical tests that are used in our experiments.

Size	Number of users	46
	Number of items	15
	Number of given ratings	624
Overall Sparsity	Overall sparsity level	10%
	Average number of ratings per user	13.56
	Average number of ratings per item	41.6
Sparsity in movies domain	sparsity level	10.65%
	Average number of ratings per user	8.9
	Average number of ratings per movie	41.1
Sparsity in books domain	sparsity level	7.39%
	Average number of ratings per user	4.6
	Average number of ratings per book	42.6
Rating Distribution	Number of ratings of likes	327
	Number of ratings of dislikes	297

Table 12. The dataset size and sparsity information

5.1.1. Ratings Predictions Metrics

Many RSs attempt to predict a user's numeric rating. For example, in the movie domain (e.g., Netflix) the RS provides the user with the likelihood that they will favor a movie, in a range of one to five stars. In these cases predictive accuracy metrics are applied, which take the distance from the predicted numeric rating to the actual rating into account. Two metrics are widely used to measure the ratings predictions: Root Mean Squared Error (RMSE) as in Eq. 3, and Mean Absolute Error (MAE) as in Eq. 4. In these equations, (u, i) refers to a user-item pair, and T is the test dataset with a predicted rating of $\hat{r}_{ui}$, and real rating of r_{ui}. Unlike MAE, large errors are not tolerated in RMSE [4]. MAE would prefer a system that makes few errors even if they are big errors (e.g., to predict a rating as 1 when in fact it is 5) whereas RMSE would prefer a system that makes many small errors rather than few big errors.

Eq. 3 $$RMSE = \sqrt{\frac{1}{|T|} \sum_{(u,i)\epsilon T} (\hat{r}_{ui} - r_{ui})^2}$$

Eq. 4 $$MAE = \sqrt{\frac{1}{|T|} \sum_{(u,i)\epsilon T} |\hat{r}_{ui} - r_{ui}|}$$

5.1.2. Usage Accuracy Metric

The metrics of usage prediction are suitable for evaluating RSs when the aim is to classify the items as interesting or non-interesting to the user. The RSs goal here is not to show the user a numeric rating, but to recommend good items. To evaluate our approach, we used the popular information retrieval system metrics of 'precision' and 'recall'. Many research papers have employed these to evaluate recommender systems, including [60], [61], and [62]. Precision indicates the proportion of recommended items that are relevant, as shown in Eq. 5. Recall is the proportion of all relevant items that are recommended, as in Eq. 6. Liked items are considered relevant, and disliked items non-relevant. There is a trade-off between precision and recall, in that the higher the number of recommended items, the more likely the user will get a non-relevant recommendation [63].

Eq. 5 $$Precision = \frac{\textbf{true positives}}{\textbf{true positives} + \textbf{false positives}} = \frac{relevant\ items\ recommended}{all\ recommended\ items}$$

Eq. 6 $$Recall = \frac{\textbf{true positives}}{\textbf{true positives} + \textbf{false negatives}} = \frac{relevant\ items\ recommended}{all\ relevant\ items}$$

Similar to precision and recall, ROC (receiver operating characteristic) curve-based metrics can successfully measure the performance of a binary classification system. ROC curves measures the proportion of the recommended non-relevant items. The resulting ROC curve can be summarized in one value, called the area underneath the ROC curve (AUC). The ROC area value represents the probability that the system can distinguish between good and bad recommendations. It reflects the value of true positives (sensitivity or recall) at multiple levels of false positives (fall-out). The ROC area is useful for finding items that are good for the user, without ranking them on the list of recommendations [63].

While precision and recall focus on the percentage of recommended relevant items, the ROC metrics determine the percentage of recommended non-relevant items. Deciding which metrics (precision and recall, or ROC curve metrics) are appropriate for evaluating a system depends on the type of system. For example, in a movie RS we might care about how relevant the recommendations are, thus precision and recall are most effective. However, the ROC curve is more suitable for a marketing RS that predicts the preferences of users who will receive products via the mail, since the cost of recommendations of items that are non-relevant to the user is expensive because they may not like the product and return it [63]. In our experiments we reported all the previously mentioned metrics.

5.1.3. Statistical Tests

The use of statistical tests helps avoid situations where the algorithm performs very well on only a specific test-set. To conduct the tests, a significance level or p-value is applied, with the p-value representing the likelihood that results are coincidental. The null hypothesis is that the performance of one algorithm is not better than the performance of another, and if the p-value is greater than a defined value, the null hypothesis is not accepted. Researchers typically define the threshold as p = 0.05 (95% confidence), though some use p=0.01, particularly when errors are very costly. Independent test cases are required before conducting a statistical test. In RSs, users are considered as test cases [4].

We applied the Wilcoxon Matched Pairs Signed Rank statistical test, which is preferred over the paired Student's t-test when the number of test cases is small and the results are not normally distributed. This test calculates and ranks the absolute differences (di) between the results of two algorithms. It then sums the ranks of positive differences ($R+$) if the result of the first algorithm is smaller than the second, and the negative ones ($R-$) when the first algorithm's result is greater than the second. When $di=0$, ranks are divided in half and added to $R+$ and $R-$ and if the number of ranks is odd, one rank is neglected. Eq. 7 shows how z is calculated, where $T = min(R+,R-)$ and N is the number of datasets (test cases). If the selected significance level is 0.05, the z score needs to be less than -1.96 for the null hypothesis to be rejected [64]. We performed the Wilcoxon test on the results using Statistical Package for the Social Sciences (SPSS [2]), a statistical analysis tool.

Eq. 7

$$Z = \frac{T - \frac{1}{4}N(N+1)}{\sqrt{\frac{1}{24}N(N+1)(2N+1)}}$$

[2] http://www-01.ibm.com/software/analytics/spss/

5.2. Experiments

The personality profile of each user was calculated in a spreadsheet directly after the user submitted their answers. The users who liked an item were isolated, and based on their profiles the IPPs for 15 items were constructed in two ways: average-based IPPs and proportion-based IPPs. Since the number of items was small, the IPP calculations were done with Microsoft Excel. As explained on section 4.1.2, the average-based IPPs represent each item with five attributes that reflect the average BF values for the users who like it, whereas with proportion-based IPPs, an item personality profile consists of 15 attributes that aggregate the number of fans who have high, average and low BF values.

The following settings were applied in all the experiments except the last one. The algorithm we used to develop the collaborative filtering system, regardless if it worked on ratings, average-based IPPs or proportion-based IPPs, was the K-nearest neighbours algorithm as implemented in WEKA [56], a widely used machine learning tool. Euclidean distance was used in calculating the nearest neighbors, and since the dataset was small, we performed ten-fold cross validation which means that the dataset was divided into 10 equal parts and tested 10 times. In every iteration, 90% of the dataset is considered the training set, and 10% the test set [20]. The results presented are the average of 22 test cases. In the experiments where usage accuracy metrics were applied, the ratings of 1 or 2 were transformed into 'dislike', and 3 or 4 to 'like'. Since movies received numeric ratings, their recommendations were evaluated using predictive accuracy metrics as well.

We conducted a number of experiments to determine the ability of the proposed approach to predict the preferences of users, in both single domain and multiple domains. We implemented two variations of the item-based collaborative filtering: one works on average-based IPP, and the other on proportion-based IPP. They are evaluated against regular item-based collaborative filtering that relies on the item-user matrix, as it is considered a powerful CF system that gives far more accurate results than a regular user-based CF [18]. We then needed to assess how the number of users would affect the representation of the item and, consequently, the performance of the system. So, we conducted an experiment with different numbers of reduced users. The last experiment was to determine whether the IPPs-system has the potential to enhance content-based RSs.

5.2.1. Single-Domain Recommendation

Goal

We intend to determine how the proposed system, in its variations of average-IPPs and proportion-IPPs, works with homogeneous items, by comparing them with a traditional ratings-based CF. This experiment will show how the three techniques perform in one-domain, and will be contrasted with their cross-domain performance in later experiments. It will allow us to ascertain which approach will provide the highest accuracy when making cross-domain recommendations.

Settings

In each of the 22 test cases the number of instances is ten movies, and the target class, which needs to be predicted, is one user's ratings of them. The comparison is done in two groups. The first group (single-domain predictive accuracy) evaluates their ability to predict the numeric ratings of users, and the second group (single-domain usage accuracy) works with the same users to predict whether they like or dislike a given movie. The number of neighbors was set to three for all three algorithms, as this is their optimal neighborhood size.

Results

In the following two sections, the predictive accuracy, measured by MAE and RMSE, and usage accuracy, measured by precision, recall, and the ROC area, of the single-domain recommendations is provided:

Single-Domain Predictive Accuracy

There are no significant differences in the errors generated by the three systems, and no statistical difference was found when the regular rating-based CF was compared to the IPPs-based approaches; the regular item-based CF had the lowest error rate, and the average-based IPPs CF had the highest. This demonstrates that our approach, whether it is average-based IPPs CF or proportion-based IPPs CF, is comparable to the item-based CF system in its predictive performance, and shows that they have the potential to be stand-alone systems.

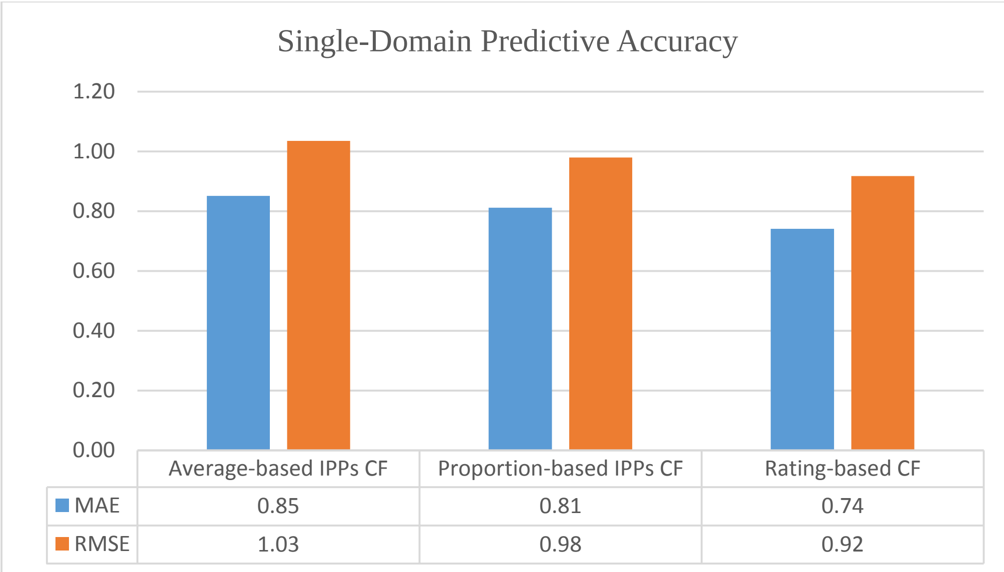


Figure 7. The MAE and RMSE generated by the three CF systems

Single-Domain Usage Accuracy

As shown in Figure 8, rating-based collaborative filtering had similar precision and recall values as proportion-based IPPs CF, although the former's ROC area value is 5 points higher than that of the latter. The average-based IPPs collaborative filtering had lower precision and ROC area, but, again, the difference is not significant. From these findings we can conclude that, though the proportion-based IPPs CF could not predict the numeric rating of a user as precisely as the regular rating-based CF, it classified the items interesting to users with the same accuracy. In addition to the predictive accuracy results, the experiment showed that the proportion-based IPPs CF performs similarly to the traditional item-based CF, and better than the average-based IPPs CF.

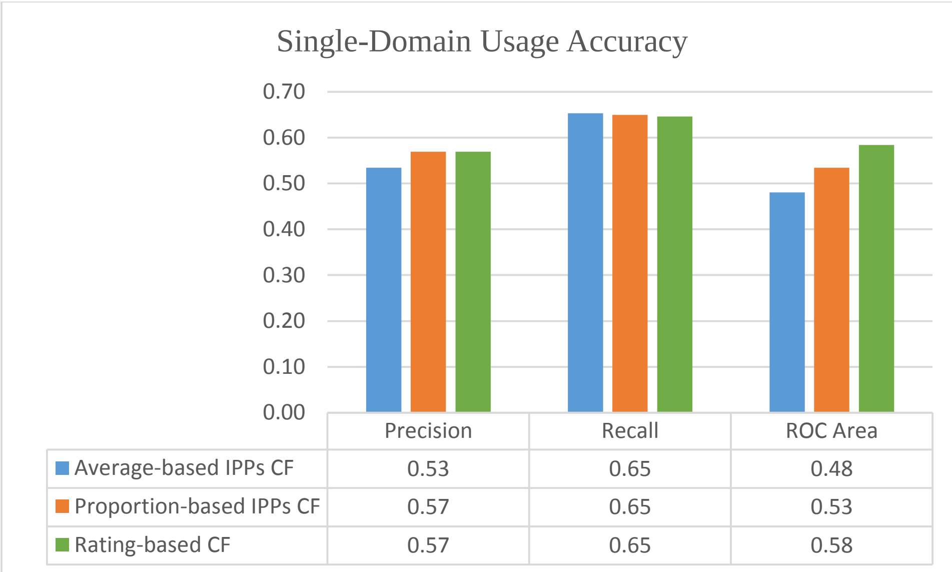


Figure 8. The difference in precision, recall and ROC area between Average-IPP-Based CF, proportion-IPP based CF and Rating-Based CF in the movies domain

5.2.2. Cross-domain Recommendations

Goal

We designed this experiment to test the quality of cross-domain recommendations given by item-based collaborative filtering, using average-based IPPs, proportion-based IPPs and ratings-based. After the addition of book ratings to the dataset, we intended to determine if introducing another domain (books) would increase or decrease the performance of the algorithms, and if the last two algorithms would have the same accuracy as in the previous experiment.

Settings

The K-nearest neighbor algorithm was given 15 items as instances to learn from, users' ratings as the target class. The same ratings as the 22 users of the previous experiment were applied. In order to compare the approaches at their best performance point, we tested the algorithms with three, five and seven neighbours. We have tried many neighborhood sizes to find which one gives the most accurate results and found that the optimal neighborhood size is five for a rating-based CF, three for a proportion-IPPs CF, and seven for average-based IPPs. The comparison was conducted with usage accuracy metrics.

Results

As Figure 9 illustrates, the proportion-based IPPs CF had the highest results, and the average-based IPPs CF the lowest. When compared to the rating-based CF, the proportion-based IPPs CF is 4, 12 and 8 points higher in precision, recall and ROC area respectively. Furthermore, cross-domain recommendations generated by the proportion-based IPP CF are 2, 10 and 16 points better than one-domain in precision, recall and ROC area respectively. As well, the rating-based CF is 2 points less in precision and recall, and 3 points higher in ROC area, when it gives cross-domain recommendations. This performance of rating-based CF is not due to a lack of rating overlap between movies and books. As shown in Table 12, the number of ratings per movie is 41.1, and per book it is 42.6. In a 46-user dataset, this is considered a high overlap. However, we found that the rating pattern in movies is different than in books; movies generally received 56% positive ratings and 44% negative ratings, while books received 45% likes and 55% dislikes. Thus, the classifier has a higher likelihood of learning the disliked books than the preferred ones. We believe that relying on the proportion-based items' personality profiles introduces another dimension to the similarity between items, and results in higher accuracy in cross-domain RSs.

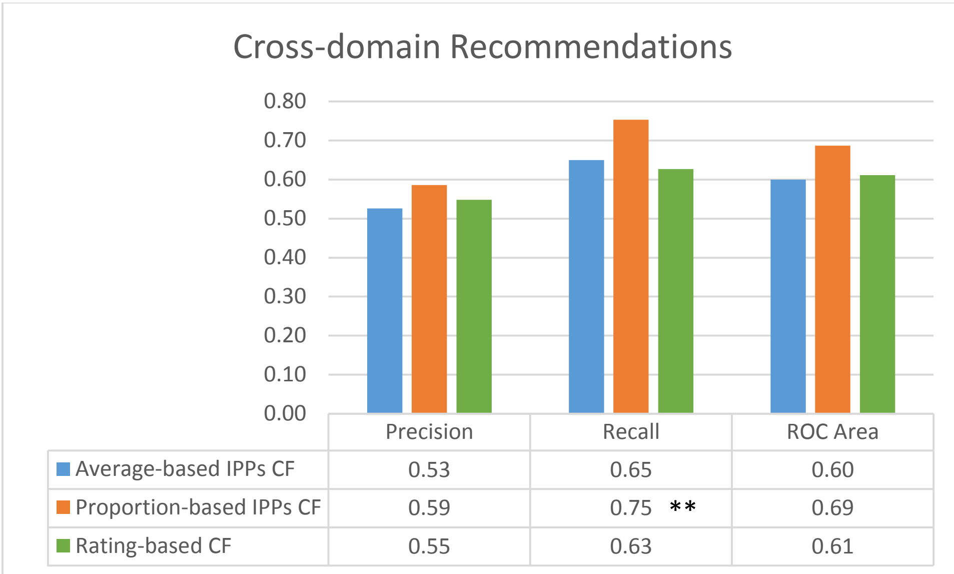


Figure 9. The difference in precision, recall and ROC areal between the three CF approaches in the multiple domains

*** indicates statistical difference at significance level 0.04*

5.2.3. Cross-Domain Recommendations with Reduced Number of Users

Goal

Since items' personality profiles (IPPs) are derived from users' profiles, they are affected by the number and types of users in the system. In a deployed system, if an IPP was developed for an item and a number of new users rated it, the IPP needs to be updated. However, the new users might have contrasting personalities, which could result in dramatic change in the representation of the IPP. In this experiment, we wanted to determine the effect of a reduction in the number of users on the performance of the three approaches. Thus, we assumed that some users do not exist and their records have been deleted. The objective was to see how the system will handle changes to the IPPs. The adjustment in IPPs means the algorithm could find different neighbours, which would result in new recommendations. For example, if the last 20 users had high Openness and Extraversion, removing their records and not including them in the IPPs could result in some IPPs having greater proportion in the “low” category in Openness and Extraversion. This would cause the algorithm to find items with a similar representation, and recommend these to the user.

Settings

The experimental settings were the same as those used in the previous cross-domain recommendations, but the average-based IPPs and proportion-based IPPs were redeveloped after deleting the last 10 users of the dataset in one experiment, and the last 20 users in the other. The same users were also excluded from the item-user matrix, in order to compare the rating-based CF to the IPPs-based approaches. In the first group of experiments with 10 reduced users, the best neighbour size for IPPs-based approaches is 3, and for rating-based CF it is 5. In the second group, all performed best at k=3.

Results

After the reduction of users, most of the IPPs were slightly different than the originals. Table 13 and Table 14 show the proportion-based and average-based IPPs for one movie, after the deletion of the last 10 and 20 users from the dataset. The change is not dramatic for all items, except the book ‘The Hunger Games’, as shown in Table 15 and Table 16.

Proportion-based IPPs	E - low	E-ave	E-high	A-low	A-ave	A-high	C-low	C-ave	C-high	N-low	N-ave	N-high	O-low	O-ave	O-high
23 users' personality profiles	0.35	0.39	0.26	0.09	0.61	0.3	0.26	0.39	0.35	0.26	0.43	0.3	0.17	0.3	0.52
21 users' personality profiles	0.33	0.43	0.24	0.10	0.67	0.24	0.29	0.38	0.33	0.29	0.43	0.29	0.19	0.33	0.48
12 users' personality profiles	0.42	0.42	0.17	0.08	0.58	0.33	0.33	0.33	0.33	**0.08**	0.42	0.**50**	0.25	0.33	0.42

Table 13. The proportion-based IPPs for the movie 'Finding Neverland' made with different numbers of users

Average-based IPPs	E	A	C	N	O
23 users' personality profiles	47.5	51.88	51.23	50.68	53.91
21 users' personality profiles	47.57	50.95	50.73	50	53.61
12 users' personality profiles	45.94	51.95	50.16	**53.91**	52.5

Table 14. The average-based IPPs for the movie 'Finding Neverland' made with different numbers of users

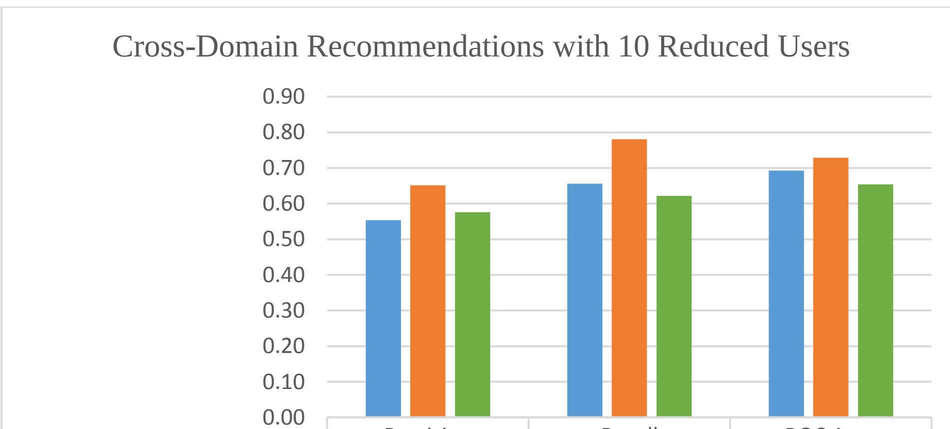
Proportion-based IPPs	E - low	E-ave	E-high	A-low	A-ave	A-high	C-low	C-ave	C-high	N-low	N-ave	N-high	O-low	O-ave	O-high
12 users' personality profiles	0.25	0.42	0.33	0.17	0.58	0.25	0.25	0.5	0.25	0.17	0.5	0.33	0.25	0.08	0.67
9 users' personality profiles	0.33	0.33	0.33	0.11	0.67	0.22	0.22	0.44	0.33	0.22	0.44	0.33	0.11	0.11	0.78
3 users' personality profiles	**1.00**	**0.00**	**0.00**	0.33	0.67	**0.00**	**0.67**	**0.00**	0.33	**0.00**	0.33	0.67	0.33	0.00	0.67

Table 15. The proportion-based IPPs for the book 'The Hunger Games' made with different numbers of users

Average-based IPPs	E	A	C	N	O
12 users' personality profiles	50.11	49.48	50.56	51.3	54.05
9 users' personality profiles	49.58444	50.24778	51.65667	50.34889	56.50667
3 users' personality profiles	**39.69**	45.93	48.30	**55.73**	51.43

Table 16. The average-based IPPs for the book 'The Hunger Games' made with different numbers of users

Surprisingly, the proportion-based IPPs CF maintained its place as a better performing algorithm in cross-domains when compared with regular CF, as shown in Figure 10 and Figure 11. When ten users are reduced, the precision, recall and ROC area are 6, 3 and 4 points better than the last experiment. The precision and ROC area declined slightly when more than ten users were removed. Similar to the proportion-based IPPs CF, the accuracy of the rating-based CF is better with ten less users, but declines with twenty less. However, the fewer users included in the average-based IPPs, the better its accuracy; it jumped from 0.55, 0.66 and 0.69 in Figure 10 when ten are removed, to 0.63, 0.84 and 0.72 in Figure 8 when more than ten are removed.



	Precision	Recall	ROC Area
Average-based IPPs CF	0.55	0.66	0.69
Proportion-based IPPs CF	0.65 *	0.78 **	0.73
Rating-based CF	0.58	0.62	0.65

Figure 10. The accuracy of cross-domain recommendations with the three approaches with 10 less users

** indicates statistical difference at significance level 0.04 and ** at significance level 0.02*

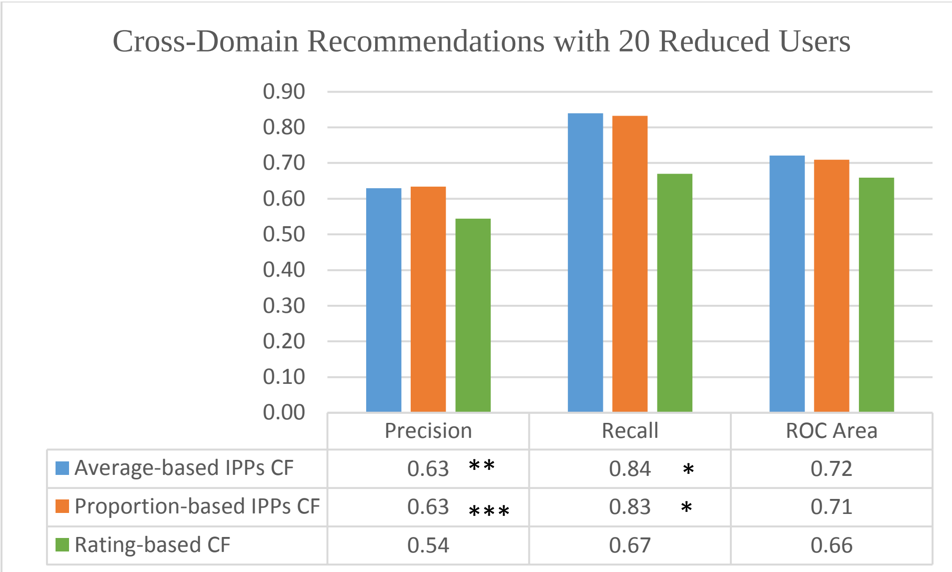


Figure 11. The accuracy of cross-domain recommendations with the three approaches with 20 less users

** indicates statistical difference at significance level 0.01, ** at significance level .03 and *** at significance level .04*

5.2.4. Enriching Content-Based Recommender with Proportion-Based IPPs

Goal

Proportion-based IPPs had the highest accuracy in the previous cross-domain results, so it would be useful to determine the effect of adding them to content-based RSs; that is, learn if they can provide a solution for low accuracy CBs.

Settings

We compared two approaches: a regular content-based RS both with and without proportion-based IPPs. The movie content was collected from IMDB[3], and comprised information about movie directors, writers, stars, plot keywords, genres, countries and release year. As mentioned, the movies were chosen because they have many common attributes. The book content was comprised of title, author, genre and publication year, as collected from Amazon[4] and Google Books[5]. The attributes the domains have in common are writers, genre and release year. We conducted two groups of experiments on the same 22-user dataset. Only movies were involved in the first experiment so the number of instances was ten, while the second experiment had 15 instances. We performed tokenization on the content of both, in that text was converted to separate words. The items were then represented in a vector space model, which weighted the terms by their term frequency (*freq)* as explained in section 2.4.1. After many trials, we found that support vector machines (SVMs), which are normally used for text-classification, provided the best results among other algorithms. The preprocessing and text classification tasks took place in WEKA, and we performed the Wilcoxon test for each group, to determine the difference between the regular CB and the CB plus IPPs.

Results

As expected, the addition of the proportion-based IPPs helped improve the performance of the content-based RS. And the performance of the cross-domain content-based, with and without the IPPs, is better than the one-domain recommendations. We concluded that a proportion-based IPPs helps improve CB accuracy.

[3] http://www.imdb.com/
[4] http://www.amazon.com/
[5] http://books.google.com/

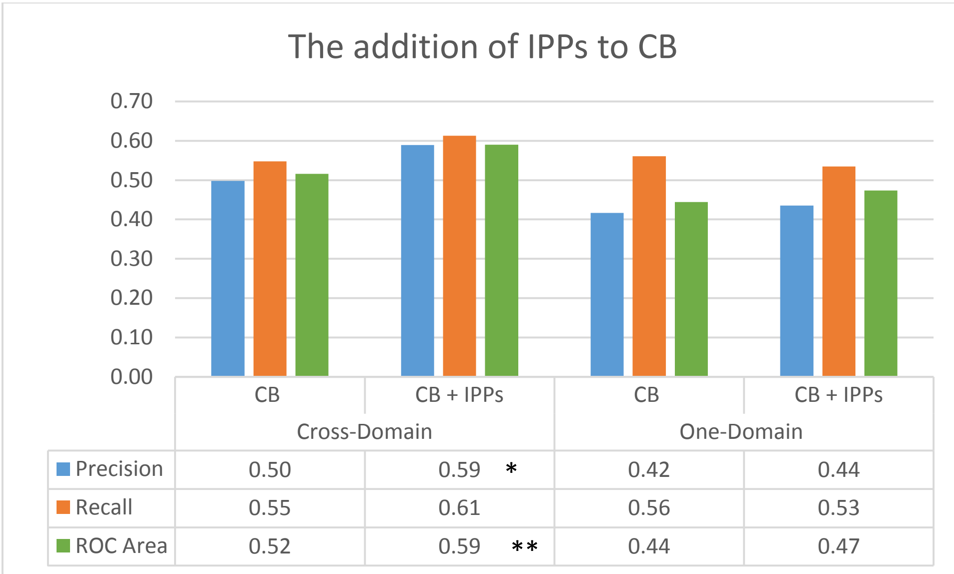


Figure 12. The performance of content-based recommender with and without proportion-based IPPs

**indicates statistical difference at significance level 0.02 and **at significance level .05*

6. Discussion and Conclusion

In this thesis, we investigated the use of customers' psychological information when making recommendations, particularly across domains, as we believe personality traits can be a suitable framework for building bridges between domains. Before conducting the research we hypothesized that, since some researchers [6] [7] found a relationship between users and their preferences, we could develop a personality profile for each item. An item's personality profile captures the Big Five traits of all users who like the item, in two techniques. The first is via average-based IPPs, which determine the mean value of the BF traits of each item's fans. In all the experiments, these IPPs did not achieve more precise recommendations than the second technique, proportion-based IPPs. As its name suggests, proportion-based IPPs find the proportion of fans in 15 categories of BF traits, and reflect the various personalities of one item's fans. Proportion-based IPPs CF achieved the best performance in almost all the cross-domain related experiments, and the performance was similar to a regular item-based CF in single-domain. For these reasons, we adopted the proportion-based IPPs CF as our proposed approach.

Collaborative filtering can only provide cross-domain recommendations when there is an overlap between the domains, but though the overlap was high in our experiments, the resulting accuracy was less than the proposed system. Content-based RSs are capable of making cross-domain recommendations when there is a common description between items of different types. After testing an RS that exploits the content of movies and books, we found great improvement when proportion-based IPPs are added to a cross-domain CB system. In addition, the pure proportion-based IPPs system performed much better than the content-based RS.

However, the conducted experiments used books and movies domains which are considered close domains. Thus, we cannot conclude at this point that the proposed system would be as effective when it works on non-similar domains. We have chosen the domains of books and movies because their overlap allows implementing a content-based RS. Some research such as [2] and [65] considered movies' genres as the domains because of the difficulty of finding an available dataset with multiple domains. Other research including ours have referred to the type of products as domains (e.g., books vs. movies). Similarly to our research, a cross-domain collaborative filtering systems [49] has used the two domains: movies and books.

6.1. Advantages and Disadvantages

In addition to the high accuracy of the system as discussed, the proportion-based IPPs CF system has the following advantages:

1. Its cross-domain recommendations do not principally depend on the overlap of ratings between items; that is, if two products were rated by distinct groups of users, the system could still find them similar if their fans have similar personalities. In contrast, traditional CF cannot provide cross-domain recommendations when no overlap is found between items.
2. It does not require prior knowledge engineering to find relationships between domains, or need commonalities in the content of items (e.g., horror movie and book).
3. It creates item personality profiles based solely on the positive ratings of users. This could be very useful, particularly in e-commerce where we can learn implicitly that a user likes a product when they view it, purchase it or read it, but we cannot determine if they dislike it.
4. It focuses on users' personalities, and thus finds deeper associations between users and items. In [6], the relationships between the preferred genres of users and their BF traits were investigated, and it was determined that test-takers who scored high in E and O, and low in

N, tend to like both horror movies and country music. However, without knowing this, one would likely not recommend a horror movie to a person who likes country music. Thus, such findings show how using personality in RSs can identify implied relationships between items.

5. It only has a small matrix that must be processed in real time, which saves time and resources. The size of the matrix will not increase when new users join the community, and it will only grow vertically if the profiles of new items are added, since each IPP vector consists of just 15 attributes. In contrast, the item-user matrix of rating-based CF increases whenever a new user or item is added, which makes it difficult for regular CF to address a large number of domains.
6. We believe relying on personality allows our system to provide users with recommendations that are diverse, noble and serendipitous; in [4], the authors identified these as factors of a successful RS. In general, most cross-domain RSs need to provide these factors; however, we believe our system surpasses the other approaches in accomplishing this. For example, a CF system could recommend an e-book to someone who bought an e-book reader, or a laptop cover to a user who purchased a laptop. Although these suggestions are from multiple domains they are not serendipitous, as the user would expect them. With cross-domain CB systems, a user who likes romance movies would not be surprised to receive a romantic song or book. However, our system could suggest a laptop cover to a user based on a song they liked, because both items were mostly liked by extroverts.
7. It employs an unusual and original method of recommendation that users might find interesting. In [41], users favor and reuse personality-based RSs more frequently than traditional rating based RSs.

The proposed system also has limitations:

1. The need to acquire users' personality information could require additional resources. For our implementation, we chose a questionnaire with 44 questions to obtain users' Big Five traits. However, personality questionnaires do not always have this many questions; in other work [16], researchers used ten question inventories. In addition, as mentioned in section 2.6.4, there are other ways to acquire personality information, such as text-based. One can use implicit methods to free users from completing questionnaires, such as analyzing their Facebook and Twitter accounts to learn about their personalities. Other research is studying using textual information generated in social media by users, to acquire their BF personality traits [66] [67]. A scenario could be getting a user's permission to observe their social media feeds, then using them to learn their BF.
2. The process of developing users' and items' profiles is not as simple as with a regular CF, as it requires significant time and effort. However, developing them is done offline, and will not affect the speed of real-time recommendations.
3. To have a personality profile developed, an item must receive enough positive ratings. Thus, similar to traditional CF, this system also encounters the new item problem.
4. Like regular CF and CB, an IPPs-based CF system requires a user to have already rated some items, before they can receive recommendations.

6.2. Future work

1. The proportion-based IPPs systems can be examined using bigger dataset that includes more users and items from different domains (e.g., clothes and electronics).

2. The text available on users’ social media accounts can be exploited to obtain the users’ BF traits as discussed in section 2.6.4.
3. Fine-grained personality acquisition could be conducted, in order to not only learn BF traits, but users’ facets as well. This would allow the IPPs to capture more user psychological information. However, fine-grained questionnaires are long and time consuming.
4. In the same way the IPPs were developed, a demographic profile of each item could also be created; it might include the ratio of females to males, for example. In addition, demographic information, particularly gender, could be added to an item personality profile, as well as users’ emotions.
5. The IPPs-based system can be developed in an adoptive manner, that updates the IPPs in real-time, and provides users with new recommendations.

7. References

[1] F. Ricci, L. Rokach and B. Shapira, "Introduction to Recommender Systems Handbook," in *Recommender Systems Handbook*, New York, Springer US, 2011, pp. 1-35.

[2] P. Winoto and T. Tang, "If You Like the Devil Wears Prada the Book, Will You also Enjoy the Devil Wears Prada the Movie? A Study of Cross-Domain Recommendations," *New Generation Computing,* vol. 26, no. 3, pp. 209-225, 2008.

[3] I. Fernadez-Tobis, I. Cantador, M. Kaminskas and F. Ricci, "Cross-Domain Recommender Systems: A Survey of the State of the Art," in *Proceedings of the 2nd Spanish Conference on Information Retrieval*, Madrid, 2012.

[4] G. Shani and A. Gunawardana, "Evaluating recommendation systems," in *Recommender systems handbook*, Berlin, Springer, 2011, pp. 257-298.

[5] G. Linden, B. Smith and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," *Internet Computing IEEE,* vol. 7, no. 1, pp. 76-80, 2003.

[6] I. Cantador, I. Fernández-tobías, A. Bellogin, M. Kosinski and D. Stillwell, "Relating Personality Types with User Preferences in Multiple Entertainment Domains," in *Proceedings of the 1st Workshop on Emotions and Personality in Personalized Services (EMPIRE 2013), at the 21st Conference on User Modeling, Adaptation and Personalization (UMAP 2013)*, Rome, 2013.

[7] P. J. Rentfrow and S. D. Gosling, "The Do Re Mi's of Everyday Life: The Structure and Personality. Correlates of Music Preferences," *Journal of Personality and Social Psychology,* vol. 84, no. 6, p. 1236–1256, 2003.

[8] T. A. Judge, D. Heller and . M. K. Mount, "Five-Factor Model of Personality and Job Satisfaction: A Meta-Analysis," *Journal of Applied Psychology,* vol. 87, no. 3, p. 530–541, 2002.

[9] R. R. McCrae and O. P. John, "An Introduction to the Five-Factor Model and Its Applications," *Journal of Personality,* vol. 60, no. 2, pp. 175-215, 1992.

[10] H. Alharthi and T. Tran, "Item-Based Collaborative Filtering Using the Big Five Personality Traits," in *Proceedings of the ACM RecSys-2014 Workshop on Recommender Systems for Television and Online Video*, Silicon Valley, 2014.

[11] H. Alharthi and T. Tran, "The Use of Items Personality Profiles in Recommender Systems," *In submission to Journal of Artificial Intelligence Research (JAIR),* June 2014.

[12] K. Musial, "Recommender System for Online Social Network," Master Thesis, School of Engineering, Blekinge Institute Technology, Ronneby, 2006.

[13] P. Lops, M. d. Gemmis and G. Semeraro, "Content-based Recommender Systems: State of the Art

and Trends," in *Recommender Systems Handbook*, Springer US, 2011, pp. 73-105.

[14] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *User Modeling and User-Adapted Interaction,* vol. 12, no. 4, pp. 331-370, 2002.

[15] X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," *Advances in Artificial Intelligence,* vol. 2009, no. 4, pp. 2-2, 2009.

[16] S. K. Lam and J. Riedl, "Shilling recommender systems for fun and profit," in *Proceedings of the 13th International World Wide Web Conference (WWW '04)*, New York, 2004 .

[17] J. Wang, A. P. de Vries and M. J. T. Reinders, "Unifying Userbased and Itembased Collaborative Filtering Approaches by Similarity Fusion," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, Seattle, WA, USA, 2006.

[18] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Item-based Collaborative Filtering Recommendation Algorithms," in *Proceedings of the 10th international conference on World Wide Web*, Hong Kong, 2001.

[19] M. J. Pazzani and D. Billsus, "Content-Based Recommendation Systems," in *The Adaptive Web*, Springer Berlin Heidelberg, 2007, pp. 325-341.

[20] J. Han, M. Kamber and J. Pei, Data Mining: Concepts and Techniques, 2 ed., The Morgan Kaufmann Series in Data Management Systems, 2006.

[21] T. Fletcher, "Support Vector Machines Explained," 2009. [Online]. Available: http://www.cs.ucl.ac.uk/staff/T.Fletcher/. [Accessed August 2014].

[22] M. Hiralall, "Recommender systems for e-shops," Business Mathematics and Informatics paper, Vrije Universiteit, Amsterdam, 2011.

[23] D. S. Darai, S. Singh and S. Biswas, "Knowledge Engineering-an overview," *International Journal of Computer Science and Information Technologies,* vol. 1, no. 4, pp. 230-234, 2010.

[24] T. Tran, "Designing Recommender Systems for E-Commerce: An Integration Approach," in *Proceedings of the 8th international conference on Electronic commerce*, New York, 2006.

[25] F. Lorenzi and F. Ricci, "Case-based Recommender Systems: A Unifying View," in *Proceedings of the 2003 international conference on Intelligent Techniques for Web Personalization*, Acapulco, Springer Verlag, 2005, pp. 89-113.

[26] J. Tang, X. Hu and H. Liu, "Social recommendation: a review," *Social Network Analysis and Mining,* vol. 3, no. 4, pp. 1113-1133, 2013.

[27] M. A. S. N. Nunes, "Recommender System Based on Personality Traits," PhD thesis, Université Montpellier 2, 2008.

[28] D. C. Funder, The Personality Puzzle, 2nd ed., New York: Norton, 2001.

[29] F. Mairesse, M. A. Walker, M. R. Mehl and R. K. Moore, "Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text," *Journal of Artificial Intelligence Research,* vol. 30, pp. 457-500, 2007.

[30] M. A. S. N. Nunes, J. S. Bezerra and A. A. d. Oliveira, "Personalityml: A Markup Language to Standardize the User Personality in Recommender Systems," *GEINTEC- gestão, inovação e tecnologias,* vol. 2, pp. 255-273, 2012.

[31] C. J. Soto, O. P. John, J. Eng, T. English, C. J. Garvan, S. Gorchoff and L. P. Naumann, "Ten Facet Scales for the Big Five Inventory: Convergence with NEO PI-R Facets, Self-peer Agreement, and Discriminant Validity," *Journal of Research in Personality,* vol. 43, p. 84–90, 2009.

[32] H. S. Sullivan, Interpersonal Theory Of Psychiatry, New York: Norton, 1953.

[33] O. P. John, L. P. Naumann and C. J. Soto, "Paradigm Shift to the Integrative Big-Five Trait Taxonomy: History, Measurement, and Conceptual Issues," in *Handbook of Personality: Theory and Research*, New York, Guilford Press, 2008, pp. 114-158.

[34] L. R. Goldberg, J. A. Johnson, H. W. Eber, R. Hogan, M. C. Ashton, C. R. Cloninger and H. G. Gough, "The international personality item pool and the future of public-domain personality measures," *Journal of Research in Personality,* vol. 40, p. 84–96, 2006.

[35] M. Dennis, J. Masthoff and C. Mellish, "The Quest for Validated Personality Trait Stories," in *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, Portugal, 2012.

[36] M. B. Donnellan, F. L. Oswald, B. M. Baird and R. E. Lucas, "The Mini-IPIP Scales: Tiny-Yet-Effective Measures of the Big Five Factors of Personality," *Psychological Assessment,* vol. 18, no. 2, p. 192–203, 2006.

[37] J. Filho and O. Freire, "On the Equalization of Keystroke Time Histograms," *Pattern Recognition Letters,* vol. 27, no. 12, pp. 440-1446, 2006.

[38] I. A. Khan, W.-P. Brinkman, N. Fine and R. M. Hierons, "Measuring Personality from Keyboard and Mouse Use," in *Proceedings of the 15th European conference on Cognitive ergonomics: the ergonomics of cool interaction (ECCE '08)*, New York, 2008.

[39] M. Mahmoud, T. Baltrušaitis, P. Robinson and L. Riek, "3D corpus of spontaneous complex mental states," in *Affective Computing and Intelligent Interaction*, Springer Berlin Heidelberg, 2011, pp. 205-214.

[40] A. Roshchina, "TWIN: Personality-based Recommender System," Master Thesis, Department of Computing, Institute of Technology Tallaght, Dublin, 2012.

[41] R. Hu and P. Pu, "A comparative user study on rating vs. personality quiz based preference

elicitation methods," in *Proceedings of the 14th international conference on Intelligent user interfaces*, Sanibel Island, 2009.

[42] "What to Rent!," [Online]. Available: http://whattorent.com/theory.php. [Accessed 25 2 2014].

[43] "MovieLens," GroupLens Research, [Online]. Available: http://movielens.org. [Accessed 2 2014].

[44] R. Hu and P. Pu, "A Study on User Perception of Personality-Based Recommender Systems," in *Proceedings of the 18th International Conference on User Modeling, Adaptation, and Personalization*, Big Island, 2010.

[45] R. Hu and P. Pu, "Enhancing collaborative filtering systems with personality information," in *Proceedings of the Fifth ACM Conference on Recommender systems (RecSys '11)*, 2011.

[46] M. Tkalˇciˇc, M. Kunaver and J. Tasiˇc, "Personality based user similarity measure for a collaborative recommender system," in *Proceedings of the 5th Workshop on Emotion in Human-Computer Interaction - Real world challenges,*, Fraunhofer Verlag, 2009.

[47] M. Szomszor, H. Alani, I. Cantador, N. Shadbolt and E. P. Superior, "Semantic modelling of user interests based on cross-folksonomy analysis," in *Proceedings of the 7th International Conference on The Semantic Web*, Karlsruhe, 2008.

[48] M. Azak, "CrosSing: A framework to develop knowledge-based recommenders in cross domains," Master thesis, Middle East Technical University, 2010.

[49] B. Li, Q. Yang and X. Xue, "Can movies and books collaborate?: cross-domain collaborative filtering for sparsity reduction," in *Proceedings of the 21st international jont conference on Artifical intelligence*, Pasadena, 2009.

[50] B. Shapira, L. Rokach and S. Freilikhman, "Utilizing Facebook Single and Cross Domain Data for Recommendation Systems," *User Modeling and User-Adapted Interaction,* vol. 23, no. 2-2, pp. 211-247, 2013.

[51] O. P. John, "The Big Five Inventory," Berkeley Personality Lab, 2007. [Online]. Available: http://www.ocf.berkeley.edu/~johnlab/bfi.htm. [Accessed 15 November 2013].

[52] O. P. John, E. M. Donahue and R. L. Kentle, "The Big Five Inventory--Versions 4a and 54," unpublished, Berkeley, 1991.

[53] V. Benet-Martinez and O. P. John, "Los Cinco Grandes across cultures and ethnic groups: multitrait multimethod analyses of the Big Five in Spanish and English," *Journal of Personality and Social Psychology,* vol. 75, no. 3, pp. 729-750, 1998.

[54] M. D. Ekstrand, J. T. Riedl and J. A. Konstan, "Collaborative Filtering Recommender Systems," *Foundations and Trends in Human-Computer Interaction,* vol. 4, no. 2, pp. 81-173, 2011.

[55] G. Schroeder, M. Thiele and W. Lehner, "Setting Goals and Choosing Metrics for Recommender

System Evaluations," in *UCERSTI2 Workshop at the 5th ACM Conference on Recommender Systems*, Chicago, USA, 2011.

[56] G. Holmes, A. Donkin and I. H. Witten, "WEKA: a machine learning workbench," in *Proceedings of the Second Australia and New Zealand Conference on Intelligent Information Systems*, Brisbane, 1994.

[57] M. Kosinski, D. J. Stillwell and T. Graepel, "Private traits and attributes are predictable from digital records of human behavior," in *Proceedings of the National Academy of Sciences (PNAS)*, 2013.

[58] R. R. Mccrae and A. Terracciano, "Personality profiles of cultures: Aggregate personality traits," *Journal of Personality and Social Psychology,* vol. 89, pp. 407-425, 2005.

[59] R. Hu and P. Pu, "Using Personality Information in Collaborative Filtering for New Users," in *Proceedings of the 2010 ACM Conference on Recommender Systems*, 2010.

[60] D. Billsus and M. J. Pazzani, "Learning collaborative information filters," in *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998.

[61] C. Basu, H. Hirsh and W. Cohen , "Recommendation As Classification: Using Social and Content-based Information in Recommendation," in *Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, Madison, 1998.

[62] B. M. Sarwar, G. Karypis, J. A. Konstan and J. Riedl, "Analysis of Recommendation Algorithms for e-Commerce," in *Proceedings of the 2Nd ACM Conference on Electronic Commerce*, Minneapolis, 2000.

[63] J. L. Herlocker, J. A. Konstan, L. G. Terveen and J. T. Riedl, "Evaluating Collaborative Filtering Recommender Systems," *ACM Transactions on Information Systems,* vol. 22, no. 1, p. 5–53, 2004.

[64] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research,* vol. 7, pp. 1-31, 2006.

[65] Y. Zhang, B. Cao and Y. Yeung, "Multi-domain collaborative filtering," in *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI)*, Catalina Island, California, USA, 2010.

[66] F. Celli, "Unsupervised Personality Recognition for Social Network Sites," in *Proceedings of ICDS*, Valencia, 2012.

[67] J. Golbeck, C. Robles, M. Edmondson and K. Turner, "Predicting personality from Twitter," in *SocialCom/PASSAT*, Boston, 2011.

8.Appendices

Appendix A: Big Five Inventory

This appendix shows the questionnaire that was used in our experiments to obtain the Big Five traits of users. Information about the inventory are given in section 2.6.4.1 and 4.1.1.

Here are a number of characteristics that may or may not apply to you. For example, do you agree that you are someone who likes to *spend time with others*? Please choose a number for each statement to indicate the extent to which you agree or disagree with that statement.

Disagree strongly	Disagree a little	Neither agree nor disagree	Agree a little	Agree strongly
1	**2**	**3**	**4**	**5**

I see myself *as someone who …*

___ 1. is talkative
___ 2. tends to find fault with others
___ 3. does a thorough job
___ 4. is depressed, blue
___ 5. is original, comes up with new ideas
___ 6. is reserved
___ 7. is helpful and unselfish with others
___ 8. can be somewhat careless
___ 9. is relaxed, handles stress well
___ 10. is curious about many different things
___ 11. is full of energy
___ 12. starts quarrels with others
___ 13. is a reliable worker
___ 14. can be tense
___ 15. is ingenious, a deep thinker
___ 16. generates a lot of enthusiasm
___ 23. tends to be lazy
___ 24. is emotionally stable, not easily upset
___ 25. is inventive
___ 26. has an assertive personality
___ 27. can be cold and aloof
___ 28. perseveres until the task is finished
___ 29. can be moody
___ 30. values artistic, aesthetic experiences
___ 31. is sometimes shy, inhibited
___ 32. is considerate and kind to almost everyone
___ 33. does things efficiently
___ 34. remains calm in tense situations
___ 35. prefers work that is routine
___ 36. is outgoing, sociable
___ 37. is sometimes rude to others
___ 38. makes plans and follows through with them

___ 17. has a forgiving nature
___ 18. tends to be disorganized
___ 19. worries a lot
___ 20. has an active imagination
___ 21. tends to be quiet
___ 22. is generally trusting

Please check: Did you write a number in front of each statement?

___ 39. gets nervous easily
___ 40. likes to reflect, play with ideas
___ 41. has few artistic interests
___ 42. likes to cooperate with others
___ 43. is easily distracted
___ 44. is sophisticated in art, music, or literature

Big Five Inventory Scoring Key [6]

Extraversion: 1, 6R [7],11, 16, 21R, 26, 31R, 36

Agreeableness: 2R, 7, 12R, 17, 22, 27R, 32, 37R, 42

Conscientiousness: 3, 8R, 13, 18R, 23R, 28, 33, 38, 43R

Neuroticism: 4, 9R, 14, 19, 24R, 29, 34R, 39

Openness: 5, 10, 15, 20, 25, 30, 35R, 40, 41R, 44

Calculate Total Scores for each of the five scales above (after appropriately reversing item scores) and enter them in the spaces below. Then calculate T-scores for each of the scales, following the formulae provided.

Total Scores Converted to T-Scores
Self-Ratings

Extraversion ___. Total Score divided by 8 = ____ (X). X minus 3.2 = ____ (Y). Y divided by 0.8 = (Z) = ____. (Z * 10) + 50 = ____ (T)
Agreeableness ___. Total Score divided by 9 = ____ (X). X minus 3.8 = ____ (Y). Y divided by 0.6 = (Z) = ____. (Z * 10) + 50 = ____ (T)
Conscientiousness ___. Total Score divided by 9 = ____ (X). X minus 3.6 = ____ (Y). Y divided by 0.7 = (Z) = ____. (Z * 10) + 50 = ____ (T)
Neuroticism ___. Total Score divided by 8 = ____ (X). X minus 3.0 = ____ (Y). Y divided by 0.8 = (Z) = ____. (Z * 10) + 50 = ____ (T)
Openness ___. Total Score divided by 10 = ____ (X). X minus 3.7 = ____ (Y). Y divided by 0.7 = (Z) = ____. (Z * 10) + 50 = ____ (T)

<u>Observer-Ratings</u>

Extraversion ___. Total Score divided by 8 = ____ (X). X minus 3.2 = ____ (Y). Y divided by 0.8 = (Z) = ____. (Z * 10) + 50 = ____ (T)
Agreeableness ___. Total Score divided by 9 = ____ (X). X minus 3.8 = ____ (Y). Y divided by 0.6 = (Z) = ____. (Z * 10) + 50 = ____ (T)

[6] Copyright Oliver P. John (1991), University of California-Berkeley, Institute for Personality and Social Research.
[7] Note that "R" denotes reverse-scored items (1=5, 2=4, 3=3, 4=2, 5=1).

Conscientiousness ___. Total Score divided by 9 = _____ (X). X minus 3.6 = _____ (Y). Y divided by 0.7 = (Z) = _____. (Z * 10) + 50 = _____ (T)
Neuroticism ___. Total Score divided by 8 = _____ (X). X minus 3.0 = _____ (Y). Y divided by 0.8 = (Z) = _____. (Z * 10) + 50 = _____ (T)
Openness ___. Total Score divided by 10 = _____ (X). X minus 3.7 = _____ (Y). Y divided by 0.7 = (Z) = _____. (Z * 10) + 50 = _____ (T)